LARGE SCALE MACHINE LEARNING FOR GEOSPATIAL PROBLEMS IN COMPUTATIONAL SUSTAINABILITY

A Dissertation
Presented to
The Academic Faculty

By

Caleb Robinson

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computational Science and Engineering

Georgia Institute of Technology

August 2020

LARGE SCALE MACHINE LEARNING FOR GEOSPATIAL PROBLEMS IN COMPUTATIONAL SUSTAINABILITY

Dissertation committee:

Dr. Bistra Dilkina, Advisor
School of Computational Science and Engineering
Georgia Institute of Technology

Dr. Duen Horng (Polo) Chau
School of Computational Science and Engineering
Georgia Institute of Technology

Dr. Chao Zhang
School of Computational Science and Engineering
Georgia Institute of Technology

Dr. Juan Moreno-Cruz
School of Environment, Enterprise and Development
University of Waterloo

Dr. Nebojsa Jojic
Senior Principal Researcher
Microsoft Research

Date Approved: April 30, 2020

To my wonderful wife, Jillian.

ACKNOWLEDGEMENTS

Thank you to my PhD advisor, Bistra Dilkina. Your guidance and teaching helped me become a researcher, then a better researcher, and now a PhD graduate. I am very thankful for all the time and patience you have spent helping me. You introduced me to computational sustainability and now I want to make it my life's work!

Thank you to my dissertation committee for the feedback, comments, and discussions over the past year.

Thank you to my labmates and friends Elias, Amrita, Aaron and all the other students at Georgia Tech and USC for the fun times and collaborations. You all have made these past five years great.

Thank you to my wife, Jillian, for putting up with me and this dissertation. Your support has been incredible.

Thank you to my parents for starting me down this path and the inspiration you both provide. I am deeply grateful for everything you have done for me.

Finally, a shout out to my siblings, Nathan, Hannah, and Elizabeth. You guys are awesome.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The UN laid out 17 Sustainable Development Goals as part of the "The 2030 Agenda for Sustainable Development", during its general assembly in 2015. Each goal consists of broad targets - such as increasing the percentage of forested land (indicator 15.1.1) - for the world to work towards in order to achieve a more sustainable future. Part of achieving these goals involves determining how to actually measure the progress that is being made towards them. Measurements of progress are necessary in order to ensure accountability, determine where resources are most needed, and weigh the effectiveness of existing sustainability efforts. However, many of the targets/indicators are prohibitively difficult to measure at global scales without algorithmic support, e.g. "the percentage of forested land" can not be evaluated at scale by human labeling efforts alone. My dissertation studies different ways machine learning methods can be used with geospatial data at large scales in order to support sustainable development efforts. I develop machine learning models and methods to address the unique computational challenges that arise in a variety of application areas - land cover mapping, human population density estimation, and human migration estimation. These challenges include training deep learning models for land cover mapping with inputs and labels from different spatial resolutions, improving the spatial generalization of these models, unsupervised training of deep learning models to act as feature extractors, training machine learning models with a domain specific non-decomposable loss function and super resolution based loss function, and creating formal couplings of human migration and sea level rise models. Our resulting models can be applied cheaply at scale, and we create the first US-wise 1m meter resolution land cover map, estimate US-wide human population from satelite imagery, and project population distributions in the United States under various sea level rise scenarios up to 2100.

CHAPTER 1
INTRODUCTION

Geospatial data is data that has a specific location on the surface of the earth. This type of data exists in many forms - for example: satellite imagery, land cover maps, census data, call detail records, hurricane tracks, geo-tagged tweets, and administrative zones - and can play a key role in decision making processes at all scales of human activity. The main challenge in incorporating geospatial data in decision making processes is often not the lack of data, but in how to interpret and analyse it over large spatial scales. This challange is well illustrated by the United Nation's (UN) Sustainable Development Goals:

During its General Assembly in 2015 the UN laid out 17 Sustainable Development Goals as part of the "The 2030 Agenda for Sustainable Development". Each goal contains a diverse set of *targets* for the world to work towards achieving by 2030. For example, Goal 15 - "Life on land" - is to "Protect, restore and promote sustainable use of terrestrial ecosystems, sustainably manage forests, combat desertification, and halt and reverse land degradation and halt biodiversity loss". Later, in the 2017 General Assembly, the *targets* were further defined through a set of 232 *indicators* [1]. Each indicator provides a quantity that can be measured to gauge progress towards its associated goal. For example, Target 15.3 in the "Life on land" goal is, "By 2030, combat desertification, restore degraded land and soil, including land affected by desertification, drought and floods, and strive to achieve a land degradation-neutral world" with the associated Indicator 15.3.1, "Proportion of land that is degraded over total land area". These indicators are global in nature, which suggests the need for large scale *geospatial data* collection and analysis. However, even with concentrated global efforts, some indicators are prohibitively expensive to measure at large scales without algorithmic assistance. The example above, "Proportion of land that is degraded over total land area", is such an indicator - the land area of the earth covers

$\sim 148,940,000 km^2$ resulting in $\sim 595,800,000$ patches of land at a $500m$ resolution. The Terra and Aqua satellites (containing the MODIS instruments) image this amount of land at a $500m$ resolution *every other day* resulting in over a billion pixels of information a day. This stream of data is impossible for humans to label or interpret directly, even with semi-automated methods, however algorithmic methods can summarize it into products such as the MODIS global yearly land cover estimates[1] which then *can* be easily used to measure global quantities like "Proportion of land that is degraded over total land area". This mapping, from multi-spectral satellite imagery to a summarized land cover data product is not trivial at a global scale, and indeed, the estimated accuracy of various layers of the MODIS global yearly land cover estimates range from 67% to 87%.

Similar stories - of needing algorithmic assistance in large scale decision making settings - exist starring other forms of *geospatial data* and in other decision making contexts. For example: the US Census records large amounts of socioeconomic data over areas the size of individual city blocks at the highest resolutions which is then used in diverse ways including business planning, economic indicators, and public health studies; sea level rise projections are mapped on to existing coastlines to help policy makers plan for different climate change scenarios; and conservation agencies examine high-resolution land cover data when planning ecosystem preservation projects.

This dissertation develops machine learning methodology that can be used to facilitate various decision making endeavors that rely on geospatial data over large scales and applies them in various settings.

1.1 Dissertation summary

This dissertation is partitioned into *technical contributions* where we propose new methods for training neural networks with geospatial data in several different *application areas*.

On the technical front we develop methodology for training neural network models with

[1] Specifically, the MCD12Q1.006 MODIS Land Cover Type Yearly Global 500m product.

auxiliary data sources and multi-resolution data (section 2.2), with humans-in-the-loop in an active learning framework (section 2.3), without labeled data using a domain agnostic self-supervised pretext task (section 2.4), and using global loss function (section 3.2) as well as a simple method for learning to aggregate gridded model predictions over larger spatial zones (section 2.5).

On the application front we have investigated estimating high-resolution land cover from aerial/satellite imagery [2, 3] (section 2.2, section 2.3), estimating human population density from satellite imagery [4] (section 2.5), predicting human migrations between administrative zones and countries [5] (section 3.2), and projecting human population distribution in the US under different sea level rise scenarios (section 3.3).

1.2 Contribution summaries

1.2.1 Training neural networks with auxiliary data sources and multi-resolution data

Our first contributions enables convolutional neural network models to be trained with larger amounts of geospatial data than possible with standard methods, which in turn improves model performance and ability to generalize spatially in the land cover mapping application. In this problem setting we have high-resolution source imagery for a wide geographic area, high-resolution labels for a limited subset of this area, and *auxillary low-resolution labels* over the entire area. This setting is consistent with many other geospatial learning problems (and problems from other application domains such as medical imagery analysis); human-labelers will carefully label objects or classes in specific source imagery from a specific location, however the same objects/classes need to be identified from other source imagery and in other locations.

Our proposed *super resolution* loss function (subsection 2.2.1) uses the joint distribution between low-resolution and high-resolution labels to train models to predict land cover at high-resolution using high-resolution source imagery but low-resolution labels. This training process can be seen as one of domain adaptation as models that are trained us-

ing *only the high-resolution labels* fail to generalize over the entire geographic area, while models trained using the super resolution loss are able to generalize. We further propose *multi-resolution data fusion* techniques that combine different sources of input imagery over multiple resolutions and time points with the available high-resolution labels during training. Low-resolution inputs give larger spatial context to models, and inputs from different points in time allow the model to see large amounts of variance due to different imaging conditions.

1.2.2 Training neural networks with humans-in-the-loop

While our first technical contribution examined the question of, "how to train deep learning models that can generalize spatially using auxiliary labeled data?", our approach relies on the existence of auxiliary labels (and the corresponding joint distribution with the high-resolution labels). To account for situations where this data is not available, we examine the question of "how to fine-tune deep learning models by soliciting few human provided labels?". Specifically, we develop a novel tool, where users can fine-tune existing land cover mapping models in a web-interface by iteratively providing labeled examples to the model, retraining the model, and observing the effects of retraining live in the web interface. This interaction lets the users build a mental model of the fine-tuning process and more efficiently provide labels than they could in a static labeling environment. We model this interaction in an active learning framework, compare labels chosen by humans to those chosen under traditional active learning criteria, and find that humans are more efficient with respect to final model performance versus time spent labeling.

Additionally, by bringing humans directly into the model training loop, versus a static setup in which a model is trained offline by a third-party, we enable them to encode problem specific information in a limited manner. Consider the variety of end-users of land cover data: some may need to use a fine-grained classification scheme, others may require certain levels of model accuracy in a specfic area, while others may need a one vs. all classifier for

a particular class of interest. These diverse needs are not always captured by existing land cover data sources, or existing model training pipelines. Our proposed framework let the end-users directly communicate these needs by iteratively training the model through a web interface instead, where existing workflows would involve a slower feedback loop in which the end-user must iterate over model design indirectly with some sort of data scientist (an intern, graduate student, data scientist, research scientist, etc.).

1.2.3 Self-supervised training of neural networks with geospatial data

We further expand on our first and second contributions by examining the problem of, "how to learn useful feature representations from unlabeled remotely sensed imagery?" Our second research question, "how to fine-tune deep learning models by soliciting few human provided labels?" assumes the existence of some deep learning model that can be fine-tuned on some task, however, deep learing models traditionally require training on large data sets to produce useful results. Practically, we previously created an interactive tool that allows users to fine-tune an existing deep learning model to new imagery, however, "what if there isn't an existing model?'. In this contribution we examine how to create such a model in a data-poor setting - specifically, one in which there are no existing labels that can be used for training a model.

Given source imagery, we want to learn a feature extractor that can be easily fine-tuned to a down-stream geospatial learning tasks. This unsupervised feature learning task is especially prescient in the geospatial imagery domain as geospatial data has many modes of variation that make *reusing* existing ML models difficult. First, imagery from a single given satellite can vary based on: location of the image, time of day, time of year, atmospheric conditions (e.g. clouds), and geolocation errors (e.g. the error in geolocation of a specific pixel may be many meters). Furthermore, imagery from different satellites (ignoring differences in time/location of the imagery) can vary based on: spatial resolution (e.g. 10 meters/px versus 1m/px), spectral resolution (i.e. which bands of light are captured,

and which wavelengths of light are captured within each band), and radiometric resolution (e.g. the values channel may be recorded at different bit-depths). Second, image *labels* are of course created using a certain source image, therefore, labeled geospatial datasets are usually specific to a narrow subset of potential variation. Machine learning models can overfit in this space when trained with such a dataset without additional efforts.

We tackle this problem by defining a novel self-supervised *pretext task*. Given an unlabeled dataset, we create pseudo-labels for each sample in the dataset by passing them through a set of randomly initialized CNNs. The randomly intialized CNNs are well defined functions each of which divides the dataset into some number of pseudo-classes. While the pseudo-classes will not necessarily have assosciated semantic labels, they will be internally coherent as each image in a peudo-class must share atleast some similar input features with the other samples in the same class. Furthermore, as the randomly intialized CNNs exactly use convolution operations to map the input images to psuedo-labels, the resulting labels will depend on the presence of spatial features in the input. We use the pairs of images and pseudo labels to train a CNN as if in a supervised multi-label setting and find that the features learned during this self-supervised training phase are transferable to downstream tasks that use 'real' labels. This method is competitive with

1.2.4 Aggregating spatial predictions with zone-level supervision

This contribution involves methodology for aggregating predictions made by geospatial deep learning models while incorporating zone level supervision, particularly in the *application* of estimating human population from satellite imagery. In section 2.5 we describe methodology for estimating gridded human population density from satellite imagery using CNNs. We disaggregate best available population count data from the US census uniformly by area onto a $\sim$1km grid of *cells* and train a CNN to predict these values from satellite imagery covering the same *cells*[2]. We test the temporal generalization of our trained mod-

[2] As these cells are of uniform area, we are predicting population density.

els by predicting population counts from an unseen year of satellite data and comparing the predictions to known ground truth data. Specifically, as ground truth population data only exists at *zone* levels, we *aggregate* our gridded predictions over US counties by summing predicted populations at the cell level for each cell in a county, then compare these summed predictions to known county population counts from the US census. We show that instead of aggregating predictions by simply summing up the predicted population values, we can use the trained CNN model as a feature extractor, sum the extracted features from all cells in a county, then learn an aggregation meta-model to predict county level population. This aggregation model is trained on zone level data from the same year that the gridded training data comes from, and intuitively serves to learn lower-resolution patterns present in the population distribution over counties. This methodology for learning an aggregation function based on zone level labeled data is generalizable to new settings, and at the most abstract level, is similar to the idea of learning set functions proposed in "Deep Sets" [6]. Aggregating cell level predictions with a learned function, versus by simply summing, has been applied in other cases where policy makers need zone-level information over familiar administrative boundaries (e.g. counties), but where the relevant geospatial models make predictions at the grid-level [7, 8, 9]. Here, if a differentiable aggregation model is used, then both the core predictive model and the aggregation model can be trained in an end-to-end fashion through labels given at the zone level.

1.2.5 A global loss function for training migration models

Finally, my last contribution is methodology for training deep learning models using the "Common Part of Commuters" (CPC) metric defined in human mobility literature [10, 11]. In section 3.2 we train machine learning models to estimate the number of human migrations between pairs of counties over the US and pairs of countries over the world. Specifically, these models use socioeconomic features from the origin and destination zones (e.g. population of origin, population of destination), as well as features between the zones

(e.g. distance) to predict the number of migrants that will move from the origin to the destination over the year that the data was recorded. These models are fit with historical data, and evaluated on held out years of data; similar to our human population estimation task, this setup tests whether the models generalize temporally. The main evaluation metric, CPC, quantifies the degree of fit between predicted and ground truth migrations given data from *all pairs* of zones. Following the principal of empirical risk minimization, we propose to directly optimize neural networks with a CPC based loss function - the quantity that is measured in application scenarios - instead of decomposable regression losses such as mean squared error. The CPC formulation is fully differentiable and thus can be used to directly train networks with, however as there are n^2 pairs of zones for n given zones in a dataset, it is impractical to train a network using CPC computed over the entire dataset (e.g. due to GPU memory constraints). We find that training models using the approximate CPC computed over large minibatches provides good empirical results, and that the size of the minibatch must be large. This methodology allows us to develop networks that outperform previously proposed human migration models.

1.2.6 A general framework for combining human migration and sea level rise models

Outside of deep learning, I detail a framework for coupling human migration and sea level rise models in section 3.3. Human migration models describe the flow of migrants between administrative areas (e.g. US counties or countries in the world) as a function of features of the origin, features of the destination, and joint features between the origin and destination. Sea level rise models describe the spatial area that will be innundated, or frequently flooded under different amount of sea level rise (usually with a time horizon, or climate conditions that will accompany different amounts of sea level rise). The direct effects of sea level rise are obvious in coastal areas - without mitigation efforts, some currently inhabited land will become uninhabited, the extent of which is directly described by sea level rise models. However, this spatial extent is not the total of the effects of sea level rise.

Displaced populations must adapt by chosing other locations to live, i.e. migrate. The sum of the decisions made by these displaced populations will change the population landscape, causing farther reaching effects. We describe a formal framework for connecting arbitrary human migration models with sea level rise models and show how this framework can be instantiated in the United States to make population projections to 2100. We also propose a method for measuring the *indirect effects*, the number of additional expected incoming migrants as a fraction of population in a climate scenario compared to a baseline scenario. Generally, such coupled models are necessary to inform policy makers as they hope to understand possible consequences of climate change and effects of adaptation and mitigation strategies.

1.3 Impacts

Throughout the work included in this dissertation we have released source code and data for replicating and extending our results. These artifacts include:

- A training/testing framework for land cover models used in section 2.2
  `https://github.com/calebrob6/land-cover`.
- An interactive web tool for incorporating human labelers in the land cover model fine-tuning loop used in section 2.3
  `https://github.com/microsoft/landcover`.
- A dataset for high-resolution land cover mapping
  `http://lila.science/datasets/chesapeakelandcover`
- Full code and data for implementing our general framework for coupling human migration and sea level rise models in the United States
  `https://github.com/calebrob6/migration-slr`.
- Code for downloading data used to estimate human population density from satellite

imagery

`https://github.com/deeppop`.

- A training/testing framework for self-supervised models used in section 2.4

  `https://github.com/calebrob/random-functions`.

On the land cover mapping front we created the first first high-resolution US-wide 1m land cover map when previously the highest resolution was at a 30m resolution. It has been made publicly available online through an API hosted by Microsoft's AI for Earth program[3], and have collaborated with groups at the World Bank to provide land cover maps in their programs, discussed on the World Bank's blog[4]. The work in this dissertation relies heavily on the 1m resolution land cover dataset for the Chesapeake Bay Watershed created by the Chesapeake Conservancy, and we are in a continued collaboration with them to update this dataset for the 2021 release they are planning. This dataset is important to many local stakeholders and agencies in the Chesapeake Bay Watershed to evaluate policy decisions that impact the overall health of the Chesapeake Bay Watershed.

Our work creating a general framework for coupling human migration and sea level rise models was reported on through several articles in news outlets such as Science Daily, TheHill, Ars Technica, Forbes, Business Insider, One Zero, and Fast Company. This has allowed our work to reach a general audience, and we have spent time over the past months answering questions from city planners and others about how to interpret the potential impacts of sea level rise.

1.4 List of publications

This dissertation consists of work from the following peer reviewed articles:

1. **Caleb Robinson**, Bistra Dilkina, Jeffrey Hubbs, Wenwen Zhang, Subhrajit Guhathakurta, Marilyn Brown, and Ram Pendyala. "Machine learning approaches for estimating

[3] Link to API definition.
[4] Link to post.

commercial building energy consumption." Applied Energy. 2017.

2. **Caleb Robinson**, Fred Hohman, and Bistra Dilkina. "A deep learning approach for population estimation from satellite imagery." In Proceedings of the 1st ACM SIGSPATIAL Workshop on Geospatial (GeoHumanities). 2017.

3. **Caleb Robinson**, Le Hou, Kolya Malkin, Rachel Soobitsky, Jacob Czawlytko, Bistra Dilkina, and Nebojsa Jojic. "Large scale high-resolution land cover mapping with multi-resolution data." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2019.

4. Kolya Malkin, **Caleb Robinson**, Le Hou, Rachel Soobitsky, Jacob Czawlytko, Dimitris Samaras, Joel Saltz, Lucas Joppa, and Nebojsa Jojic. "Label super-resolution networks." In Proceedings of the 7th International Conference on Learning Representations (ICLR). 2018.

5. **Caleb Robinson**, and Bistra Dilkina. "A machine learning approach to modeling human migration." In Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies (COMPASS). 2018.

6. **Caleb Robinson**, Bistra Dilkina, and Juan Moreno-Cruz. "A modeling framework for studying migration patterns under climate change with an application to sea level rise." PLOS One. 2020.

7. **Caleb Robinson**, Anthony Ortiz, Kolya Malkin, Blake Elias, Andi Peng, Dan Morris, Bistra Dilkina, and Nebojsa Jojic. "Human-Machine Collaboration for Fast Land Cover Mapping." In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). 2020.

CHAPTER 2

DEEP LEARNING IN REMOTE SENSING APPLICATIONS

2.1 Introduction

Land cover data and gridded human population estimates are two of many types of geospatial data that can benefit decision making for sustainable purposes. In this chapter we examine how to infer such data over large scales from remotely sensed satellite and aerial imagery with varying levels of labeled data.

2.1.1 Land cover mapping

Land cover data - a classification of the Earth's surface into categories based on physical material such as: water, forest, low vegetation, and impervious surfaces - is a type of geospatial data that is important component in many sustainability-related efforts. The UN has estimated that, "approximately 20% of the SDG indicators can be interpreted and measured either through direct use of geospatial data itself or through integration with statistical data" [12]. For example, the following indicators include land cover data either directly or as a secondary source [13]:

2.4.1 Proportion of agricultural area under productive and sustainable agriculture

6.3.2 Proportion of bodies of water with good ambient water quality

9.1.1 Proportion of the rural population who live within 2 km of an all-season road

11.1.1 Proportion of urban population living in slums, informal settlements or inadequate housing

11.3.1 Ratio of land consumption rate to population growth rate

11.7.1 Average share of the built-up area of cities that is open space for public use for all, by sex, age and persons with disabilities

15.1.1 Forest area as a proportion of total land area

15.1.2 Proportion of important sites for terrestrial and freshwater biodiversity that are covered by protected areas, by ecosystem type

15.3.1 Proportion of land that is degraded over total land area

15.4.1 Coverage by protected areas of important sites for mountain biodiversity

15.4.2 Mountain Green Cover Index

Conservation biologists can use land cover data to target the creation of riparian forest buffers - areas that protect streams from collecting pollutants from adjacent areas - an important conservation task that can improve the overall "health" of a watershed. As a specific example, the Chesapeake Bay Conservancy uses, "flow path data and high-resolution land cover data to identify opportunity areas for planting riparian forest buffers within a specified distance of the flow paths. Once these restoration opportunity areas (ROAs) are identified, they can be characterized by the land cover composition and modeled sediment and nutrient loading of the upstream land area that drains through the ROA" [14]. Land cover data is also useful for monitoring drivers of climate change and in predictive modeling of potential climate change futures [15].

In spite of its usefulness in sustainability applications, high-resolution land cover data is difficult to collect at scale. Land cover data is a product of satellite or aerial imagery - it must be generated through a combination of modeling and human labeling efforts from a more "raw" form of geospatial data. In standard GIS workflows, the process of generating land cover data is traditionally a semi-automated one whereby models are used to make initial land cover predictions from imagery, then, in a laborious process, human experts are used to correct the output of the model [16].

In section 2.2 we frame land cover mapping as an image segmentation problem and develop computational methods for scaling land cover models over large areas. In section 2.3 we develop a human-in-the-loop framework for fine-tuning land cover models in an online

setting. Finally, in section 2.4 we develop a self-supervised method for training good feature extractor models without labeled data such that they can serve as a good starting point for fine-tuning to land cover and other related tasks.

2.1.2 Human population density estimation

Many countries around the world conduct censuses to gather rich information about their population's size, composition, and demographics. While these censuses only happen every 5 to 10 years depending on the country, they are highly important for government policymakers and planners. In the United States sub-national *population estimates* based on census data are used extensively. County level population estimates are used in: "federal and state funds allocation", "denominators for vital rates and per capita time series", "survey controls", "administrative planning and marketing guidance", and "descriptive and analytical studies", according to Long, 1996 [17]. According to the US General Accounting Office, more than "70 federal programs distribute tens of billions of dollars annually on the basis of population estimates", and "[e]ven more money was distributed indirectly on the basis of indicators which used population estimates for denominators or controls" [17]. *Population projections* rely on census based *population estimates* to fill data gaps since a previous census was taken. These can be used to project the consequences of long-term health effects such as aging and infectious diseases such as HIV/AIDS. Traditionally, population projections rely on the interaction between three factors: fertility, mortality, and migration[1]. To project population characteristics at a future date, demographers make assumptions about fertility and mortality in a current population and further assume how many people will move into or out of an area before that date, i.e., migration. But since population projections carry inherent uncertainty, demographers often times can use previous projections and projection errors to better inform future projections.

Unfortunately, censuses in many countries are non-representative due to limited civil

[1]Public Reference Bureau: `http://www.prb.org/Publications/Reports/`

registration systems and other issues [18] which make population projections less effective. Satellite and aerial imagery are collected globally however, and provide a strong signal of human population density to support population projections.

In section 2.5 we frame human population estimation from satellite imagery as an image classification problem and develop aggregation methods for producing zonal population estimates.

2.2 High-resolution land cover mapping

Land cover mapping is a semantic segmentation problem: each pixel in an aerial or satellite image must be classified into one of several land cover classes. These classes describe the surface of the earth and are typically broad categories such as "forest" or "field". High-resolution land cover data ($\leq$ 1m / pixel) is essential in many sustainability-related applications. Its uses include informing agricultural best management practices, monitoring forest change over time [19] and measuring urban sprawl [20]. However, land cover maps quickly fall out of date and must be updated as construction, erosion, and other processes act on the landscape.

In this section we identify the challenges in automatic large-scale high-resolution land cover mapping and develop methods to overcome them. As an application of our methods, we produce the first high-resolution (1m) land cover map of the contiguous United States. We have released code used for training and testing our models at `https://github.com/calebrob6/land-cover`.

Scale and cost of existing data: Manual and semi-manual land cover mapping of aerial imagery is currently expensive and scales poorly over large areas. For example, the Chesapeake Conservancy spent 10 months and $1.3 million to produce a high-resolution (1m) land cover map of the Chesapeake Bay watershed in the Northeast US. This project, the largest of its kind, labeled only $\sim$160,000 km^2, or 2% of the US [14]. Existing bench-

mark land cover segmentation datasets and studies are limited to even smaller scales. The DeepGlobe challenge dataset [21, 22] covers a total area of 1,717 km^2, the Dstl satellite imagery dataset [23] covers $\sim$400 km^2, the UC Merced land use dataset [24, 25] covers just 7 km^2, and the ISPRS Vaihingen and Potsdam dataset [26] contains fewer than 36 km^2 of labeled data. In comparison, a single layer of aerial imagery of the contiguous US covers 8 million km^2 (8 trillion pixels at 1m resolution), occupying 55 TB on disk – two orders of magnitude larger than ImageNet, a standard corpus for training computer vision models. Deep learning-based approaches for land cover mapping have shown to be effective, however, in limited-size studies: [27] compare common CNN image classification architectures at a 6.5m spatial resolution in a small part of Newfoundland, Canada, while [28] use a multi-resolution approach for handling panchromatic and multispectral bands separately at a 1.5m spatial resolution in a $\sim$4000 km^2 area.

Figure 2.1: Example 1 km^2 image patches. **Top row:** NAIP imagery from 2012, NAIP imagery from 2015, ground truth land cover. **Bottom row**: Landsat leaf-on imagery, Landsat leaf-off imagery, NLCD land cover.

Model generalization: High-resolution land cover labels at 1m resolution only exist at concentrated locations. Such localized label sets have not been successfully used to classify land cover on a larger scale. Indeed, we show that neither standard random forest approaches [29], nor common semantic segmentation networks generalize well to new ge-

ographic locations: models trained on a single Northeast US state see their performance degrade in the entire Northeast region, and further in the rest of the country. Existing GIS methodology such as Object Based Image Analysis (OBIA) [20, 30] suffers from the same generalization issues, yet costs more in terms of data and effort to deploy. For example, OBIA methods have been used to create high-resolution (1m) land cover maps in part of a county in Indiana [31] and the city of Phoenix, Ariz. [32], but rely on human-engineered features and hand-derived rule-based classification schemes.

In view of this, we develop methods for generalizing models to new regions, achieving **high-quality results in the entire US**. Specifically, we augment high-resolution imagery with low-resolution (30m) satellite images, extend labels with low-resolution land cover data that we use as weak supervision, and augment data with inputs from multiple points in time (see Fig. 2.1). We evaluate models trained with these methods in the US: a) with ground-truth labels from the Chesapeake Bay area in the Northeast US; b) through visualizing their outputs in other US regions; and c) by comparing their predictions with low-resolution land cover labels over the entire US. As low-resolution satellite and land cover data sources are widely available, such as Landsat satellite imagery, or Global Land Cover [33], our methods are applicable wherever high-resolution imagery exists.

Evaluation: An important consideration for large-scale land cover mapping tasks is the cost associated with executing a trained model over massive scales. We run our best model, a U-Net variant, over the entire contiguous US to produce a **country-wide high-resolution land cover map**. This computation took one week on a cluster of 40 K80 GPUs, at a cost of about $5000, representing massive time and cost savings over the existing methods used to produce land cover maps. We provide a web tool through which users may interact with the pre-computed results – see `http://aka.ms/cvprlandcover` – exposing over 25TB of land cover data to collaborators.

In practice, land cover models must be verified and updated with human input. Our

proposed models can be adapted to new regions with relatively little human labor. In a **study with domain experts**, we evaluated our best model (trained in the Chesapeake Bay area from the Northeast US) on a region in Iowa, then obtained manual corrections on $\sim 1\%$ of this territory. Using these corrections, we fine-tuned our model output, which reduced both the overall error and the manual labor required to perform corrections over the entire area.

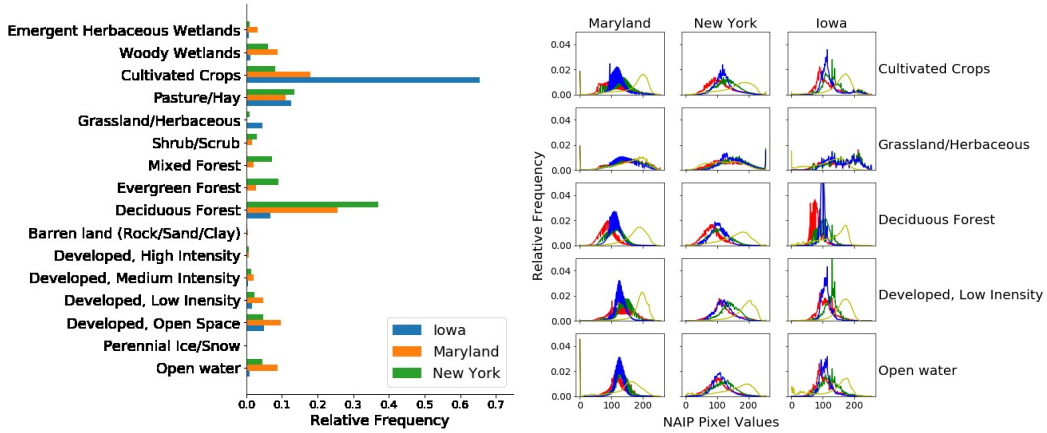


Figure 2.2: **Left** Inter-state differences in NLCD class composition, **Right** Inter-state NAIP color histograms per NLCD class. Different states cover different geographies and have different *land use* purposes (e.g., over 60% of Iowa is covered with cultivated crops, while Maryland has a more uniform distribution of land cover) and different color profiles for each class.

2.2.1 Multi-resolution data fusion

We assume that we are given a training set of pairs of high-resolution satellite or aerial imagery and high-resolution land cover labels, $\{(X^{(t)}, Y^{(t)})\}_{t=1}^{T}$ where $X^{(t)} = \{X_{ijk}\}_{i,j,k}^{(t)} \in \mathbb{R}^{h \times w \times c}$ is a multispectral image with height h, width w, and channel depth c, and $Y^{(t)} = \{Y_{ij}\}_{i,j}^{(t)} \in \{1, \ldots, L\}^{h \times w}$ are the associated land cover labels. A straightforward approach for training a deep neural network, $f(X; \theta) = \hat{Y}$, on this fully supervised semantic segmentation problem involves minimizing a standard loss function with respect to the network's parameters, i.e., $\min_\theta J(Y, \hat{Y})$. This approach generally works well in problems where potential *test* images are sampled from the same generating distribution as the train-

ing images.

This assumption does *not* hold for the land cover mapping problem as high-resolution input images will vary due to differences in: geography of the earth, atmospheric conditions (e.g., cloud cover), ground conditions (e.g., flood conditions), quality and type of sensor used in capturing the image, time of day or season that the image was captured, etc. Indeed, these differences are obvious in the input data we use later in this study, see Figure 2.2. As a result, models trained with standard deep learning methods fail to generalize over wide areas, see Section 2.2.4. We propose the following methods to improve model performance:

Low-Resolution Input Augmentation (LR): Publicly available low-resolution satellite data has been collected globally since 1972, starting with the Landsat 1 satellite. We find that augmenting high-resolution imagery with low-resolution imagery that has been averaged over large time horizons improves model performance. This averaged low-resolution imagery is less susceptible to sources of local noise that impact high-resolution imagery and can therefore be used by models to smooth such noise. Formally, for every high-resolution image, $X^{(t)}$, we assume that we can access a low-resolution image $Z^{(t)} \in \mathbb{R}^{h' \times w' \times c'}$. We resample the low-resolution imagery to the same spatial dimensions ($h \times w$) as the high-resolution imagery, then concatenate the two image sources, giving new input imagery $X'^{(t)} \in \mathbb{R}^{h \times w \times (c+c')}$.

Label Overloading (LO): The available hand-labeled land cover maps are created from a single time point of aerial imagery, but high-resolution imagery is collected periodically. Given that the true land cover of a location is not likely to change over short time scales, we augment our training dataset by pairing high-resolution training labels with high-resolution image inputs from different points in time. Specifically, given an image and labels (X, Y) at some point in time, we augment our training set with all pairs (X', Y), where X' ranges over imagery from all time points when it is available. Although this method has the potential to introduce confusion in cases where the high-resolution labels do not match

the content of the other images (due to land cover change by construction, flooding, etc.), we demonstrate that it allows models to learn invariance to spurious high-resolution image differences.

Input Color Augmentation (Color) We found that within small geographical regions, individual pixel color is a very predictive feature for land cover classification, whereas across geographical locations, color is very inconsistent for each class. As a result, models trained on limited geographical locations overfit on color. Thus, we choose to add random color augmentation to input images. Given a training image, we randomly adjust the brightness and contrast per channel by up to 5%. Specifically, given a single channel of an image, $X_c \in \mathbb{R}^{h \times w}$, and the mean pixel intensity for that channel, $\overline{X}_c$, we sample $t, b \in \mathbb{U}(0.95, 1.05)$, as the contrast and brightness adjustments, then compute the transformed image as $X'_{ijc} = t(X_{ijc} - \overline{X}_c) + b\overline{X}_c$.

Super-Resolution Loss (SR): We augment the training set with additional *low-resolution* labels from outside of the spatial extent in which we have *high-resolution* training data to better inform the model. We incorporate pairs of high-resolution imagery and low-resolution *accessory labels* corresponding to the same spatial extent as the imagery, but where low-resolution labels are assigned to larger (e.g., 30×30m) *blocks* of the image. We assume each accessory label class c determines a (known) distribution over frequencies of each high-resolution label, ℓ. We then use a variant of the super-resolution loss function of [34], which encourages the model to match its high-resolution predictions to the fixed distributions given by the low-resolution labels while favoring high certainty of predictions.

Specifically, we assume each low-resolution label c determines a distribution $p_{mean}(\ell|c)$ over the frequency of labels of high-resolution class ℓ in a block labeled c, with mean $\mu_{c,l}$ and variance $\sigma^2_{c,l}$. These parameters are computed on a small subset of labeled data where both kinds of labels are available. Alternatively, they could be manually set. We view the probabilistic output of the core segmentation model, p_{net}, as generating labels indepen-

dently at each high-resolution pixel, inducing a corresponding distribution $p_{\text{out}}(\ell|c)$ over label counts in each block. We then minimize the super-resolution loss, $\text{KL}(p_{net}\|p_{mean})$, over all blocks in the input image.

We incorporate this metric into the overall loss function by minimizing a weighted sum of the standard high-resolution loss (categorical cross-entropy) and the super-resolution loss:

$$J(Y_i, \hat{Y}_i) = \gamma(\text{HR loss}) + \eta(\text{SR loss}). \tag{2.1}$$

In offline experiments we have found that a ratio of $\gamma : \eta = 200 : 1$ balances the two losses effectively. We use this setting in all experiments in this work.

2.2.2 Data

Imagery data sources: High-resolution (1m) aerial imagery from the USDA National Agriculture Imagery Program (NAIP), and low-resolution (30m) multispectral satellite imagery from the USGS's Landsat 8 satellite.

Label data sources: High-resolution (1m) land cover labels from the Chesapeake Conservancy [14], based on imagery from the years 2013-2014, and low-resolution (30m) land cover labels from the 2011 National Land Cover Database (NLCD) [35].

Figure 2.1 shows aligned example images from each of these data sources. Combined, these datasets are ~ 165TB on disk.

We use NAIP data from 2011 to 2016, which provides 2 to 3 layers of high-resolution imagery for each location in US. This allows us to implement the **Label Overloading** method by pairing our high-resolution labels with multiple years of NAIP imagery. We implement the **Low Resolution Input Augmentation** method by creating two sets of Landsat 8 Tier 1 surface reflectance products: a median of non-cloudy pixels from 2013 to 2017 over the April-September months (leaf-on) and a similar product over the October-March months (leaf-off). These layers are both resampled to the 1m-resolution grid used in the

NAIP data.

The high-resolution land cover labels from the Chesapeake Conservancy consist of 4 land cover classes – water, forest, field, and impervious surfaces – for the Chesapeake Bay watershed, outlined in black in Figure 2.4. The low-resolution NLCD labels are from the 2011 data product and consist of 16 land cover classes covering the contiguous US. We use these low-resolution labels as additional training supervision with the **Super-Resolution** data fusion method. Each label at the 30m resolution suggests a distribution of high-resolution labels: e.g., an NLCD label "Developed, Medium Indensity" suggests on average 14% of the block is forest and 63% of the block is impervious surface. See Section 2 in the SI for more details about these correlations.

We use an additional set of high-resolution land cover labeled data in the case study in Iowa (Sec. 2.2.4), derived from multiple dates of aerial imagery and LiDAR elevation data, as a held out test set [36]. We map the 15 land cover classes in this Iowa dataset to the same 4 Chesapeake land cover classes that our model is trained on according to the Iowa class specifications.

As expected, the distribution of NLCD low-resolution classes and their appearance varies between states (see Figure 2.2 for class distributions and color histograms). In addition, there is not a standardized national method for collecting NAIP imagery: it is collected on a 3-year cycle by different contractor companies, with collection years differing between states (see Figure 2.1). These sources of variability in the NAIP imagery must be accounted for in order to build models that will generalize over the entire US using only *high-resolution* training data from the Chesapeake Bay region, motivating our study.

2.2.3 Experiments

Neural Network Models

We consider three network architectures: **FC-DenseNet**, **U-Net**, and **U-Net Large**. Each of these architectures contains the basic structure of four down-sampling and four up-

Table 2.1: Models that are trained solely on high-resolution labels generalize poorly, regardless of the choice of architecture, training, and testing sets. Compared to the results in Tab. 2.2, we see that almost all models without multi-resolution data fusion perform worse than any of the models with multi-resolution data fusion.

		North Chesapeake Test Set		**Iowa Test Set**	
Training Set	**Models**	**Accuracy**	**Jaccard**	**Accuracy**	**Jaccard**
Maryland	RF	37.11%	15.60%	74.95%	31.47%
	FC-DenseNet	71.05%	44.92%	77.87%	41.01%
	U-Net Large	78.06%	50.50%	82.31%	47.06%
	U-Net	61.19%	39.62%	79.07%	47.28%
	U-Net + Adapt	63.33%	42.55%	79.69%	44.10%
South Chesapeake	RF	41.16%	17.96%	72.33%	30.52%
	FC-DenseNet	72.46%	47.83%	74.07%	38.34%
	U-Net Large	72.38%	46.51%	61.56%	37.44%
	U-Net	59.42%	40.47%	71.00%	40.93%
	U-Net + Adapt	62.88%	41.60%	62.95%	39.28%

sampling layers. For down-sampling, we use a simple 2×2 max-pooling. For up-sampling, we use deconvolution (transposed convolution) with fixed interpolation, which is useful for reducing checkerboard artifacts [37]. The U-Net models [38] contain three convolutional layers between successive down/up-sampling modules, with batch normalization after each convolution operation and before a ReLU activation function. The FC-DenseNet model [39] instead contains "dense blocks" made up of three convolutional-batchnorm-ReLU layers. The FC-DenseNet model uses 32 filters in a convolution layer immediately after the input and 16 filters in all other convolutional layers. The U-Net model contains 64 3×3 filters in the first three convolutional layers and 32 3×3 filters in all other convolutional layers. The U-Net Large model contains 32 3×3 filters in the first three layers and double the number of filters after each pooling layer, except in the representational bottleneck layer that uses 128 filters.

For training, we use the largest minibatch that will fit in GPU memory and the RMSProp optimizer with a learning rate schedule starting at 0.001 with a factor of 10 reduction every

Table 2.2: We show the effect of our data fusion methods. (1). Regardless of the choice of models (RF, U-net), the training set (Maryland, South Chesapeake), and the testing set (North Chesapeake, Iowa), adding data fusion methods significantly improved the results. (2). Increasing model capacity, only provides diminishing accuracy and Jaccard returns. The U-Net Large model only performs slightly better than the U-Net model. (3). Our best performing models are able to generalize excellently to Iowa, with an accuracy of 93.35% and Jaccard score of 71.32%.

			North Chesapeake Test Set		**Iowa Test Set**	
Training Set	**Data Fusion Methods**	**Models**	**Accuracy**	**Jaccard**	**Accuracy**	**Jaccard**
Maryland	LR + Color	RF	64.37%	47.27%	83.03%	49.86%
	LR + Color + LO	RF	75.06%	54.57%	81.94%	49.90%
	SR	U-Net	84.72%	57.72%	80.91%	40.45%
	SR + Color	U-Net	85.11%	59.16%	86.50%	45.03%
	SR + LR + Color	U-Net	88.45%	70.90%	90.95%	62.17%
	SR + LR + Color + LO	U-Net	89.52%	74.11%	92.36%	68.91%
	SR + LR + Color + LO	FC-DenseNet	89.74%	74.30%	91.81%	68.81%
	SR + LR + Color + LO	U-Net Large	**90.31**%	75.41%	**92.93**%	70.66%
South Chesapeake	LR + Color	RF	67.15%	49.08%	88.90%	54.60%
	LR + Color + LO	RF	77.57%	53.86%	83.86%	52.89%
	SR	U-Net	86.85%	62.49%	77.83%	42.03%
	SR + Color	U-Net	87.11%	63.34%	79.71%	42.68%
	SR + LR + Color	U-Net	89.13%	72.83%	93.07%	67.66%
	SR + LR + Color + LO	U-Net	90.61%	76.29%	93.06%	71.12%
	SR + LR + Color + LO	FC-DenseNet	90.52%	76.16%	93.28%	71.17%
	SR + LR + Color + LO	U-Net Large	**90.68**%	76.60%	**93.35**%	71.32%

6000 mini-batches. We use the Python CNTK library for implementation [40].

Baseline Methods

Random forests (RF) have been used extensively in previous literature for low-resolution land cover classification [41, 42], usually with Landsat imagery, and recently for high-resolution land cover classification [29, 30]. The RF results in the high-resolution setting are promising in areas for which there are high-resolution labels, however show problems generalizing to new geographies [29]. We therefore train a baseline Random Forest model (**RF**) to predict the land cover class of a single pixel from raw pixel values of that pixel and the surrounding pixels within a given radius (in the L_∞ metric). Our offline experiments show that increasing this *feature radius* hyperparameter improves model performance slightly when no augmentation techniques are used, but does not increase perfor-

mance with Low Resolution data augmentation is used. The RF model we use has a feature radius of 1, is created with 100 trees, and uses the default parameters from the Python scikit-learn library [43] otherwise.

To improve the generalization ability of supervised models in an unsupervised fashion, domain adaptation methods [44, 45, 46, 47, 48] learn to map inputs from different domains into a unified space, such that the classification/segmentation network is able to generalize better across domains. We use an existing domain-adversarial training method [44] for the land cover mapping task (**Adapt**). In particular, we attach a 3-layer domain classification sub-network to our proposed U-Net architecture. This subnetwork takes the output of the final up-sampling layer in our U-Net model and classifies the source state (New York, Maryland, etc.) of the input image as its "domain". In addition to minimizing segmentation errors on limited image domains, we also train the segmentation network to *maximize* the error of the classification sub-network. In this way, the segmentation network learns to generate more domain-invariant features.

Model Training and Evaluation

We train all models on two sets: the state of **Maryland** and its superset, the lower half the Chesapeake Bay region (**South Chesapeake**). We test on a set consisting of the upper half of the Chesapeake Bay region (**North Chesapeake**) as well as held out land cover data from Iowa (**Iowa**). In training, we uniformly sample $\sim 100,000$ 240×240 pixel patches with high-resolution land cover labels from the training set. If Adapt or Super Resolution is used, we sample an additional $\sim 150,000$ 240×240 patches from across the US. In the Adapt case, these additional samples are without labels, while in the Super Resolution case, we include their low-resolution NLCD labels. For a given set of tile predictions, we compute the accuracy and average Jaccard index (i.e. intersection-over-union) over the four high-resolution classes.

The relationship between training on data from a single state and testing on the held out

North Chesapeake set mimics the relationship between the entire Chesapeake Bay region and the rest of the US. Maryland data is restricted both geographically (i.e., models trained there will not be able to observe features found in other parts of the Chesapeake Bay) and in observed NAIP sensor variance (i.e., all the imagery in Maryland from a given year will be collected in the same manner). A similar relationship will hold between the Chesapeake Bay region and the remainder of the US, e.g., it is impossible to observe deserts in the Chesapeake Bay, and there will be NAIP imagery conditions that are unobserved in the Chesapeake Bay region, but present in other parts of the country.

Training on South Chesapeake exposes models to more of the variation that is likely to be present in North Chesapeake, thus making the generalization easier than that of Chesapeake to the whole US. Indeed, the NLCD class composition of **South Chesapeake** is similar to that of **North Chesapeake**, but not similar to the remainder of the US.

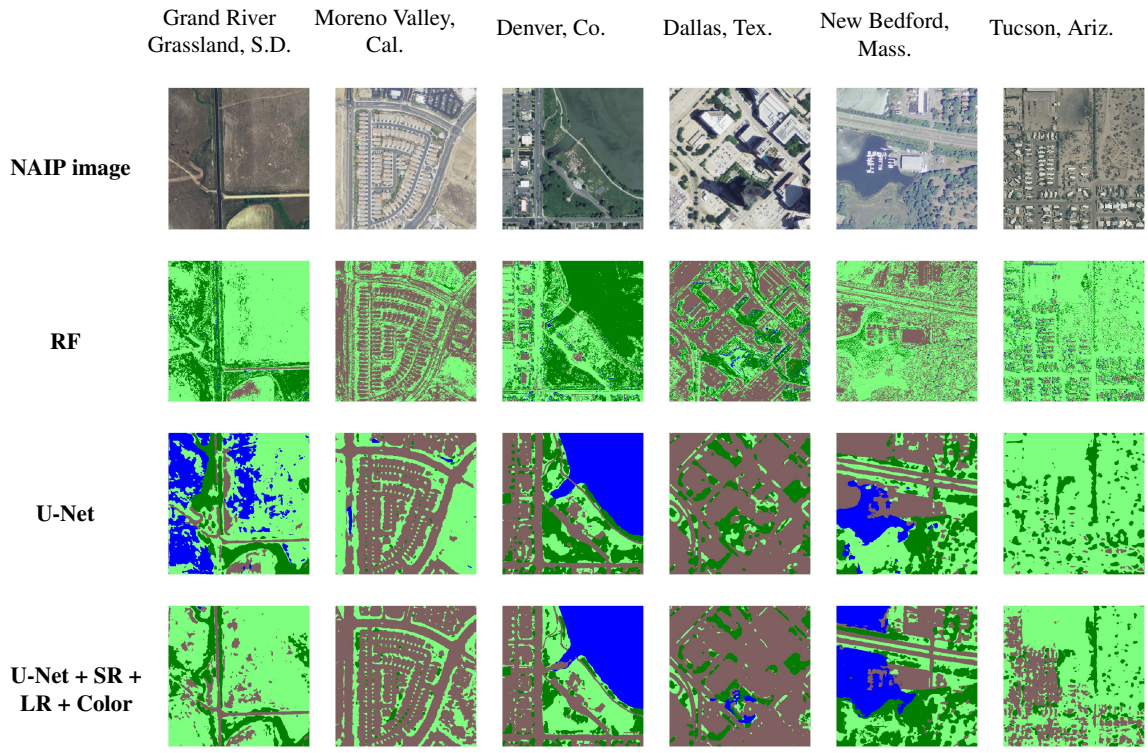


Figure 2.3: High-resolution land cover predictions, and accompanying NAIP imagery, for different models in choice locations where ground truth labels are not available. Here, the color map is the same as in the high-resolution ground truth image from Figure 2.1.

2.2.4 Results

Model Generalization

The results in Table 2.1 show the performance of our models when trained solely on high-resolution labeled data, i.e., without our multi-resolution data fusion methods. These results show that the models are not generalizing well: adding more training data (South Chesapeake vs. Maryland training sets) results in poorer performance on the Iowa test set. The models that are trained in Maryland have Jaccard scores of less than 50% on the Iowa test set, but relatively high accuracies, which suggests that they are biased towards predicting the majority class (overwhelmingly "field" in Iowa). The benefits of using higher-capacity models, like the U-Net Large, or more complex models, like the FC-DenseNet, are not expressed in this land cover mapping problem. Lastly, of note, the domain-adaptation method we use does not give a significant increase in model performance.

Table 2.2, however, shows how the progressive addition of our data fusion methods improves model performance. More specifically, for models trained in **Maryland**, each data fusion method increases the performance in both the North Chesapeake set *and* in the held out Iowa set in terms of accuracy and Jaccard scores. For models trained on South Chesapeake, the benefits of **LO** are not as prevalent as with the restricted Maryland training subset. In this case, the South Chesapeake set must contain additional features that are not present in the Maryland set before Label Overloading is used. Of note, increasing model capacity provides diminishing accuracy and Jaccard returns. The U-Net Large model only performs slightly better than the U-Net model. Our best-performing models are able to generalize excellently to Iowa, with an accuracy of 93.35% and a Jaccard score of 71.32%.

In addition to the quantitative model results, we visualize the land cover output from several of our models over a set of hand-picked scenes from locations outside of the Chesapeake Bay in Figure 2.3. We choose these locations to capture potential failure cases and to display model behaviour in interesting settings. In most locations, our best model (last

row) correctly identifies features mislabeled by the **RF** baseline and other versions of the model trained without data fusion methods. In the last column, an image from Tucson, Arizona, we observe that the two baseline models, without data augmentation, are not able to identify a collection of houses in an ambient desert. Our best-performing model in the last row is able to correctly identify the houses, but does not identify the road.

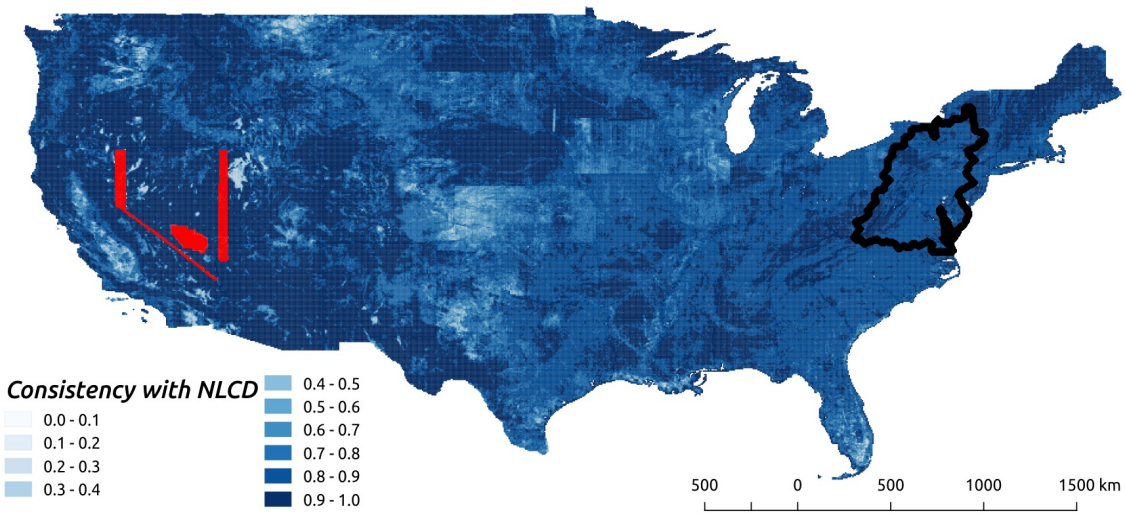
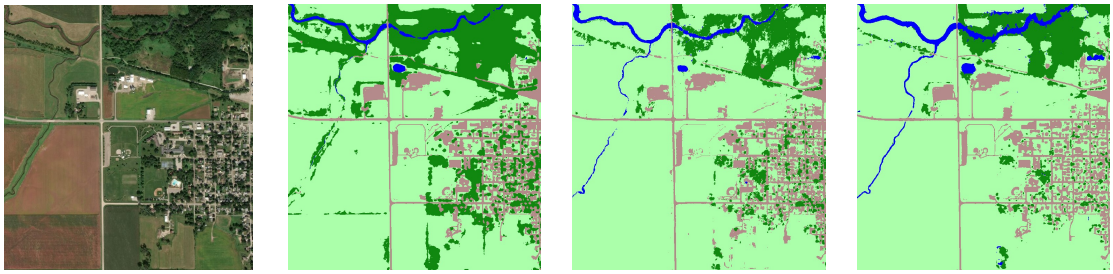


Figure 2.4: Maps showing the high-resolution *consistency with NLCD* over the entire US. Lower values represent an 'inconsistency' between our model estimates and the expected high resolution labels (using high-resolution label distributions per NLCD class from the Chesapeake Bay area). Areas for which there is no input data, or data errors prevented our model from running are shown in red.

Middle Cedar Watershed Case Study

Our partners at the Chesapeake Conservancy are working with the Iowa Agricultural Water Alliance (IAWA) to pilot new techniques to facilitate watershed management planning throughout the state of Iowa. High-resolution land cover data is important in this setting to rapidly identify specific recommendations for how to improve land management and water quality while minimizing the impact to farm operations.

Thus, we ran an early version of our model over the entire area of Middle Cedar watershed in Iowa, an area of 6260km^2, and gave the results to our partners. The partners used



(a) (b) (c) (d)

Figure 2.5: Qualitative evaluation of the land cover Quality Assurance (QA) process. Given source imagery, (**a**), and initial land cover prediction, (**b**), our collaborators used their QA methodology to correct the prediction, (**c**). Finally, we simply re-weighted the softmax outputs of our model to match the correct predictions as much as possible. This process is very time efficient (no model training). We show re-weighted results in (**d**).

their quality assurance (QA) methodology to correct the model's systematic errors over a geography that was $\sim 1.1\%$ of the area of the total watershed. This methodology involves comparing the model output with NAIP imagery, a Normalized Difference Vegetation Index (NDVI) layer, and a Normalized Difference Surface Model (nDSM) layer to identify classification errors. The NDVI and nDSM layers help to identify misclassifications in vegetation and mistakes that can be captured with height differences (e.g. low vegetation misclassified as trees) respectively. The first round of this process resulted in corrections of three broad classes of errors: incorrect prediction of the "field" class bordering roads, rounded building corners, and water values predicted in shadows. The corrections represented $\sim 2\%$ of the pixels in the evaluated geography and cost 30 hours to perform. Using this feedback, we tuned our model's per-pixel class probabilities with a global transformation to best fit the corrections and generated a new map over the entire watershed using this transformation.

Formally, we are given n corrected samples from our set of model predictions. We sample another n pixels that were not corrected in the QA process in order to balance the dataset, then form a matrix $\mathbf{X} \in \mathbb{R}^{2n \times 4}$, of our model's probabilistic output, and vector, $\mathbf{y} \in \mathbb{R}^{2n \times 4}$, of the accompanying labels (one-hot encoded). We find a transformation $\mathbf{W} \in \mathbb{R}^{4 \times 4}, \mathbf{b} \in \mathbb{R}^4$, such that $\mathbf{XW} + \mathbf{b} = \hat{\mathbf{y}}$ minimizes the categorical cross-entropy with

$\mathbf{y}$. The learned transformation, $\mathbf{W}$ and $\mathbf{b}$, can now easily be applied across any number of pixels. This method is able to correct 89.7% of the errors made by the original model, and, under the assumption that our model is making the same systematic errors across the whole testing region, is able to save the ~ 2700 hours of manual labor that would be required to correct the entire area. In Figure 5 of the SI we display the progression of this feedback process for a small patch of land in a corrected area. This method is a cheap way to incorporate domain expert feedback to a model's existing predictions, and can further be embedded in a model generation loop, where new versions of the original model are fine-tuned with samples from the broader corrected area, and updated predictions are looped back into the QA process.

US-Wide Land Cover Map

We used the approach of our best model – including all data fusion methods – to generate a full-US land cover map. For training data, we used high-resolution labels from the entire Chesapeake Bay region and low-resolution NLCD labels sampled over the entire US. The correlations between NLCD and high-resolution labels, $\mu_{n,c}$, for the Super Resolution loss were manually tuned for this model, rather than estimated from data in the state of Maryland (as in our experiments)[2].

Cost: The size and complexity of the network used to create the full-US land cover map will largely determine the cost of the operation. For example, the Dense Fusion Classmate network that won the DeepGlobe land cover mapping competition requires 8 GPUs to train and would be prohibitively costly for full-US inference [49]. The FC-DenseNet103 architecture [39], on which the Dense Fusion Classmate network is based, can fit on a single GPU but will incur an $\sim 270\%$ increase in cost over our U-Net Large model when run over the entire US. Our full-US map was generated with the U-Net Large architecture,

[2]The input to the model we trained for this purpose has a small difference compared to the best model reported in Table 2.2: we used an median of all available Landsat 8 imagery, not separating leaf-on and leaf-off months.

which only has a 19% cost increase over the U-Net and FC-DenseNet models.

Evaluation: In Section 2.2.4 we discuss a "benchmark" visualization set of patches that we use to inspect a model's performance on important terrain features, and in the SI we show a web application to interactively explore our models' predictions. However, these are not sufficient for discovering all cases where our model is performing poorly. It is prohibitively time-consuming to qualitatively evaluate the performance of our model by simply sampling patches of model input vs. predicted output. Considering this, we use the low-resolution labels to approximate the performance of our model across the entire US by computing a metric we call *consistency (of high-res labels) with NLCD*. First, we compute the high-resolution class distribution for each NLCD label, $p_{mean}(y|n) = \mu_{n,y}$, as described in the **SR** data fusion method. We let $\rho_{n,y} = \mu_{n,y} / \max_{y'} \mu_{n,y'}$, normalizing the high-resolution class means for each NLCD label by the maximum value in that distribution. Now, given a set of N high-resolution predictions, $\{y_1, \ldots, y_N\}$, and the associated NLCD labels, $\{c_1, \ldots, c_N\}$, we compute the *consistency with NLCD* value, $\lambda = \frac{1}{N} \sum_{i=1}^{N} \rho_{c_i,y_i}$. This definition can be thought of as a charitable "accuracy" score for a given set of predictions[3]. In general, the aim of this metric is to identify potential problem areas in our US-wide evaluation – areas in which the high-resolution labels *do not* have *consistency with NLCD*. Finally, we show the approximate accuracy map for this model run in Figure 2.4, with an average *consistency with NLCD* of 87.03%.

2.3 A human-in-the-loop framework for fast land cover mapping

Machine learning models are usually imagined as artificially "intelligent" agents that mimic human autonomy and generalization abilities: having explored their training environment, machine learning models are supposed to choose their actions independently and reliably

[3] As an alternative, we could define a deterministic mapping from NLCD labels to high-resolution labels and directly compute an "accuracy" surrogate, but this will heavily penalize predictions in areas where multiple high-resolution classes may occur with high frequency, such as cities. We expand on and show results for this definition in Section 3 of the SI.

in similar situations. While this notion of intelligence guides the design and testing of new algorithmic ideas, in practice, the resulting algorithms are rarely capable of either autonomy or generalization. Instead, human decision-making is present throughout a AI model's development and lifetime: researchers and engineers acquire data with a specific goal in mind, then work on finding and tuning the methods that handle the peculiarities of the data well. When the algorithm is eventually deployed, it often suffers from *domain shift*, where slight changes in the statistics of real-world input compared to the training input can degrade performance considerably. Thus, the algorithm is constantly reevaluated through human monitoring, which may trigger a process requiring repeated data acquisition and retraining [50]. Hence, most practical deployments are better thought of as examples of hybrid – rather than purely artificial – intelligence. Active learning loops can be seen as an approximate model of such hybrid human-machine intelligence, as long as humans are allowed deeper involvement than just as labeling oracles. More specifically, the hypothesis is that if humans are allowed to choose which samples to label, and subsequently *fine-tune* a deployed model with, then they will be able to correct model errors, such as those from input *domain shift*.

Image segmentation is an ideal task to test hybrid human-machine intelligence, as segmentation is a natural ability of humans [51] and one where humans can exploit the spatial structure of input to identify errors. Recent work has probed the complementary abilities of humans and machines on image labeling tasks [52, 53]. We investigate whether it is possible to maximize performance on one such application, land cover mapping from high-resolution satellite imagery, by directly integrating humans into the training loop instead of isolating the artificially intelligent component. Our methods can be applied in settings where the human-in-the-loop can quickly search and evaluate the deployed model over unlabeled examples. This is the case in geospatial image labeling tasks and medical image segmentation tasks (e.g., segmenting tumor-infiltrated lymphocytes in pathology imagery), where unlabeled points have a strong spatial structure (i.e., points can be thought of as part

of a large continuous image).

Figure 2.6: National Agriculture Imagery Program (NAIP) aerial imagery (**top row**) with modeled land cover estimates (**bottom row**). Existing supervised learning models, trained for generating land cover labels from aerial imagery, do not generalize well due to the large spatial and temporal variances in aerial imagery. Creating accurate land cover maps at a massive scale therefore requires additional human interventions. We propose an interactive model fine-tuning system, coupling human labelers and machine learning models, for facilitating these interventions.

We summarize our main contributions as follows:

- We design an *interactive web tool* that enables users to test a high-resolution land cover model on any patch of land on a satellite map, then – in the same interface – relabel pixels of their choosing and retrain ("fine-tune") the model in real time (see Fig. 2.7).

- We study the effectiveness of the *combination of different active learning query methods with different model fine-tuning methods* in an offline study and find that querying for labels at randomly selected points outperforms or nearly matches standard active learning *query methods* (see Fig. 2.8).

- In an online user study, we examine **how well human labelers function as sample *query methods* compared to automatic selection methods**. We find that humans perform significantly better, even compared to learning systems in which the model is told on which points it is making labeling errors (see Fig. 2.9).

- Furthermore, we show that **the value of human-provided labels increases with the time humans spend using the tool**: users develop a *theory of mind* of the learning system that improves the performance of the hybrid human-AI intelligence over time, and the hybrid intelligence increases users' *trust in the AI*.

2.3.1 Background

Active learning

A traditional active learning setup consists of a parameterized model, an unlabeled 'pool' of data, a data *query method* (also known as the *data selection method*), and a *labeling oracle*. One iteration of fine-tuning the model consists of 1.) utilizing the *query method* to choose data points for labeling, 2.) querying a *labeling oracle* for the labels, and 3.) fine-tuning the model parameters to these additional data samples [54].

The purpose of the *query method* is to pick unlabeled data that, when labeled, will provide the largest benefit to the model. In active learning, the learner is allowed to ask for help by querying the label oracle, but it must know which samples to request labels for. Conventional approaches ask the oracle to label instances with low prediction confidence [55, 54]. Other techniques consider the similarity between an unlabeled sample and existing labeled samples as a selection criterion [56]. Another recent approach models uncertainty in labeling oracles to improve the efficiency of active learning [57]. Meta-learning (or "learning to learn") active learning query methods rely on existing labeled datasets drawn from the same distribution as the unlabeled data pool [58, 59], and as such will not be effective when the model must be adapted to work in a shifted distribution. Finally, *query method* algorithms suffer from "unknown unknowns": a model's self-inspection does not reliably reveal what it does not model well. This is the case in most ML algorithms, including deep neural networks [60].

On the other hand, by observing the effects of their decisions on a model being retrained on-the-fly, human labelers can adapt their own data selection process to reflect not only their

understanding of the data, but also their developing intuition regarding the inner workings of the model and its adaptation algorithms.

Land cover mapping

Land cover mapping – the segmentation of aerial or satellite imagery into land cover classes such as "water", "forest", "field/low vegetation", or "impervious surface" (Fig. 2.6) – has attracted reinvigorated interest in machine learning research [2, 21, 22, 49, 61]. High-resolution land cover maps are an essential component in environmental science, agriculture, forestry [19], urban development [20], the insurance and banking industries, and for demography in developing countries [62]. Satellite imagery is being produced on an increasingly frequent basis. However, despite their importance, high-resolution land cover maps are not yet widely available as neither ML algorithms nor human labor scale appropriately [2].

To a machine learning or computer vision researcher, land cover mapping is a semantic segmentation problem. Machine learning models are not yet able to generate high-resolution (1m / pixel) land cover labels with performance that matches human labeling. A major obstacle is that high-resolution land cover labels for training such models only exist in small, specialized locations [21, 22, 24, 25]. In [2], it is shown that a state-of-the-art deep neural network trained on 1m-resolution images and labels from a much larger (160,000 km^2) dataset [14] in the Chesapeake Bay watershed (north-eastern US) still does not perform well in the mid-western US. Other recent work also utilizes additional, more broadly available input data [63, 34, 64]; however, all existing land cover models are biased by the geographic locations on which they were trained. Large systematic errors in predictions limit their applicability and are challenging to detect at scale.[4] Finally, the classification tasks are constantly shifting. While one dataset may segment vegetation simply into "low vegetation" and "tree canopy", other applications may require delineating coffee

[4]For example, imagery of the contiguous US at 1m resolution covers 8 trillion pixels.

farms from orchards.

To a Geographic Information Systems (GIS) professional, however, land cover mapping is an inherently human-driven process augmented by technology. Accurate and useful labels themselves, not a training dataset for ML algorithms, are the immediate goal. The process typically starts with color-based segmentation algorithms that create initial maps, followed by experts who provide labels in different areas, creating rules on the fly, and then manually correcting the remaining errors. The labor efficiency of the process may increase as the humans learn how to use these tools better, but is not boosted by quick adaptation of the classification algorithms themselves.[5] This makes land cover mapping at the resolution and scale needed today cost-prohibitive for most agencies.

A hybrid system for accurate and efficient land cover labeling would more tightly integrate the human and machine efforts. Here we investigate a land cover mapping workflow where users' work immediately affects the performance of prediction algorithms. Our design, which incorporates human feedback integrated in real time as training points for our model, can be seen as an instance of machine teaching [65, 66], as humans deploy their own intelligence to identify and correct mislabeled points in an effort to improve the model. However, our system does not attempt to create an autonomous entity, capable of generalizing, as the final result: the ability to efficiently label large areas is the goal, and the final trained algorithm is but one aspect of the overall workflow. To a human, the ML model is simply a powerful macro that they (re)define on the fly in order to amplify their work. To the ML model, the human is the source of data to learn from. Together, this hybrid system holds the potential to outperform existing GIS workflows as well as pure ML approaches in cost and accuracy.

[5] Typically, separately tuned random forests are used, although neural networks are rapidly gaining traction.

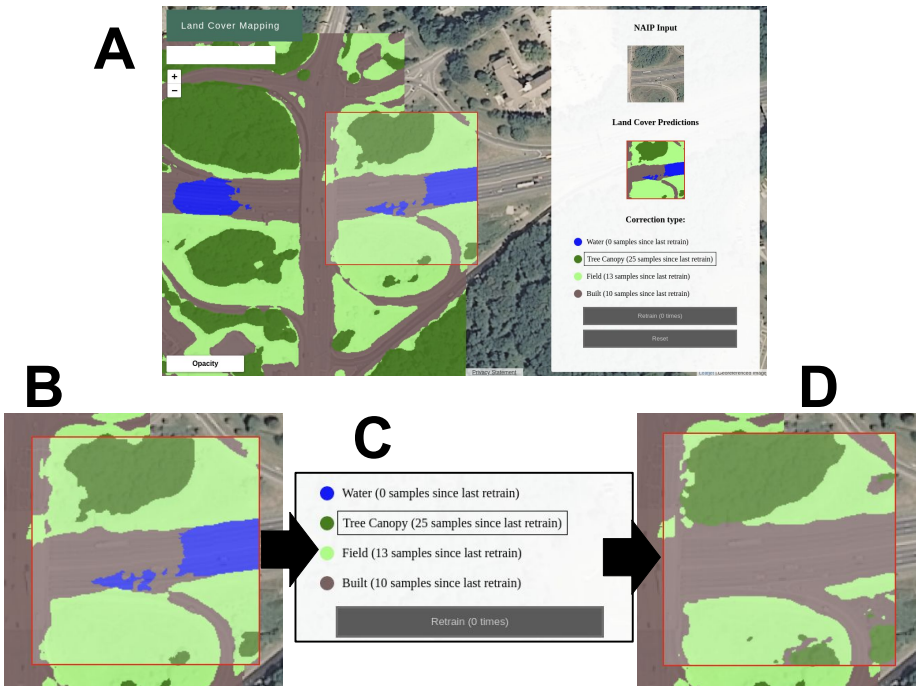


Figure 2.7: User interface of our land cover labeling tool. (See the video in the Supplementary Material) (**A**) Land cover prediction results are overlaid on top of the map. (**B**) The user can easily identify misclassified pixels and (**C**) submit corrections by clicking on the map. (**D**) Pressing "Retrain" updates the model and displays new land cover predictions in the interface. In this example, the user provided a handful of point corrections in the impervious surface initially misclassified as water.

2.3.2 Study design

We focus on the following task: given a pretrained segmentation model, which was trained on 1m-resolution imagery and a four-class land cover map of Maryland [14], we would like to quickly (within at most 15 minutes) produce accurate maps for regions of 1m-resolution imagery in New York State. This change in the geographic region where the model is to be applied represents a *domain shift*. We aim to create the map of each region by slightly changing the parameters of the Maryland model to fit a limited number of guidance points in the new areas.

We vary two parameters in our study: the **fine-tuning method** and **query method**.

The **fine-tuning method** is the algorithm for retraining the model to fit new guidance points. Such a method needs to be fast and sample-efficient. As we have ground truth

data in the entire Chesapeake watershed, including Maryland and N.Y., various choices for fine-tuning can be evaluated offline.

The **query method** is the method for selecting guidance points on which to fine-tune the model on a new region. The main object of our study is to compare automatic methods, such as random selection or active learning approaches, to **hybrid (human-guided) methods**, where users iteratively view the current model's predictions, correct the labels at points of their choice, and trigger model retraining. The traditional active learning approaches to automatic selection of points to query can also be studied offline on a fully labeled dataset (Sec. 2.3.3).

We implement the **hybrid (human-guided) method** by developing a web tool that allows users to iterate between labeling and testing the model (Fig. 2.7). The tool exploits the spatial nature of the data in the task, allowing the user to zoom and pan in the high-resolution imagery to find areas where they want to test the current algorithm. Upon a click on the map, the prediction of the current model on a surrounding 500m$\times$500m patch of land is overlaid on the map. The user can then label pixels of their choice, either where they see errors or for some other reason they think that the label will be useful. They can induce near-instant retraining of the model at any time with the click of a button. After that, they can check how well the retrained model works by clicking on the imagery again.

Base segmentation model

Our base segmentation model takes input patches of high-resolution (1m) four-band aerial imagery from the USDA National Agriculture Imagery Program (NAIP) and outputs a segmentation of the image (per-pixel classification) over four land cover classes (water, forest, field, impervious surfaces). The default training label datasets are from [14].

Formally, the model is a convolutional neural network (CNN) that produces probabilities of each class at each image pixel. Given the model parameters θ and an image $X = \{x_{ijk}\} \in \mathbb{R}^{w \times h \times c}$ (where $c = 4$ is the channel depth and $w \times h$ are the image dimen-

sions), the model outputs a probability distribution over the target classes at each pixel, i.e., $f(\theta, X) \in \mathcal{D}(n)^{w \times h}$, where $\mathcal{D}(n)$ is the probability simplex on the $n = 4$ output classes. This yields distributions over labels $P_\theta(\hat{y}_{ij}|X)$ for each coordinate (i, j).

The network is similar to the U-Net architecture [38, 22]. We trained the network on randomly selected patches of imagery and land cover labels sampled from the state of Maryland [34, 14]. The training settings match those of [34]; see the supplementary material for details.

2.3.3 Offline active learning experiments

Table 2.3: Results of fine-tuning on 400 points selected by different query methods, averaged over four target areas and five random seeds.

	Last 1 Layer		**Last 2 Layers**		**Last 3 Layers**		**Group Params**		**Dropout**	
Query method	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU
Baseline	0.725	0.510	0.725	0.510	0.725	0.510	0.725	0.510	0.725	0.510
Random	0.806	0.608	0.825	0.677	0.824	0.658	0.791	0.562	0.787	0.597
Entropy	0.736	0.501	0.731	0.587	0.765	0.572	0.760	0.520	0.741	0.550
Min-Margin	0.811	0.608	0.834	0.701	0.832	0.685	0.793	0.580	0.785	0.601
Mistakes	0.729	0.551	0.781	0.631	0.756	0.621	0.787	0.575	0.762	0.609

As discussed in Sec. 2.3.2, we investigate different methods for fine-tuning a pre-trained model and querying for new label data in a different domain. In these experiments, and in the online experiments described in Section 2.3.4, the new domain is imagery from four 84km^2 *areas* in New York. Our offline experiments are meant to identify the optimal fine-tuning and query methods, which are then used in online user studies. In our offline experiments, the base segmentation model is adapted to a small number – 10 to 2000 – of *automatically chosen* labeled pixels (less than 0.01% of each target area). Then the performance is evaluated on the entirety of the target areas.

Fine-tuning methods

The following **fine-tuning methods** were tested:

LAST k LAYERS Following [67], the final k convolutional layers in the U-net architecture have their weights exposed as trainable via gradient descent (initialized from the weights of the base model), while all other parameters in the network are held fixed. Here, $k \in \{1, 2, 3\}$.

GROUP NORMALIZATION PARAMETERS Inspired by the success of feature-wise transformations [68] in neural style transfer [69] and visual question answering [70], we extended it for model fine-tuning. Our U-net architecture uses group normalization [71] in the final convolutional layers. The group normalization parameters affect large groups of filters in each layer via a single affine transformation, with the assumption that filters within a group are correlated. Thus, training these parameters to fit new training points causes correlated changes in the layers' outputs, providing a regularized mechanism to affect the entire network, in contrast with full backpropagation, which affects all weights in the chosen layers.

DROPOUT We effect dropout, i.e., set the outputs of a fixed subset of the neurons to 0, in the final k convolutional layers. Searching for the binary mask that minimizes a loss is a discrete optimization problem, which we solve using a simple genetic algorithm. Here we use $k = 5$ and a mean dropout rate of 0.2, but we conducted only limited experiments due to the high cost of this method, which requires evaluation of the model at all sample points at each of 64 mutation iterations. However, we hypothesize that this highly constrained method is less prone to overfitting than techniques based on backpropagation.

Query Methods

Motivated by [55, 54], we also investigated three **query methods** for selecting the additional 10 to 2000 labeled pixels used by the fine-tuning methods:

RANDOM Sample points (i^*, j^*) uniformly randomly from the training area.

ENTROPY Select points which maximize the Shannon entropy of output distributions over classes:

$$(i^*, j^*) = \arg\max_{(i,j)} \left(-\sum_{\ell} P_\theta(\hat{y}_{ij} = \ell | X) \log P_\theta(\hat{y}_{ij} = \ell | X) \right). \tag{2.2}$$

MIN-MARGIN Select points which minimize the difference between probabilities assigned to the most-likely and second-most-likely classes:

$$(i^*, j^*) = \arg\min_{(i,j)} \left(P_\theta(\hat{y}_{ij} = \ell^1_{ij} | X) - P_\theta(\hat{y}_{ij} = \ell^2_{ij} | X) \right), \tag{2.3}$$

where ℓ^1_{ij} and ℓ^2_{ij} are the two most likely classes under $P_\theta(\hat{y}_{ij}|X)$.

We also include the following method, the purpose of which is to make a comparison with *humans* selecting mistake points in our online study. It is *not* an automatic query strategy, as it assumes the model has access to an all-knowing labeling oracle *before* it chooses where to query the oracle for labels. It simply imitates a teacher that feeds randomly chosen mistake points to the model.

MISTAKES Uniformly sample points (i^*, j^*) where the model's prediction disagrees with the ground truth.

Results

Because it is prohibitively costly to select points using the ENTROPY, MIN-MARGIN, and MISTAKES methods at *every* training iteration, we approximate this procedure by batching: periodically evaluating the model on the training area and selecting the optimal points among a large set of 10000 uniformly sampled locations. Namely, we evaluate the model and select a new batch of points after 10, 40, 100, 200, 400, 1000, and 2000 points have been chosen.

The experiments are repeated five times with different random seeds for each combination of the adaptation method, point selection strategy, and target area. The average adaptation performance when methods use only 400 labeled pixels – close to the number labeled by users in our online studies – is shown in Table 2.3, while the variation in accuracy across the whole range of additional training points is shown in Figure 2.8. (See also the supplementary materials for the full set of curves.)

For all fine-tuning methods, we observed a similar ranking of the performance of active learning query methods, with MIN-MARGIN performing best, but only slightly better than RANDOM, and ENTROPY performing worst. Most interestingly, the MISTAKES method performs significantly worse than RANDOM: even giving the model access to ground truth knowledge does not improve performance. On the other hand, in online experiments (Sec. 2.3.4), we show that replacing this mock "uniform teacher" with a human teacher *does* improve performance.

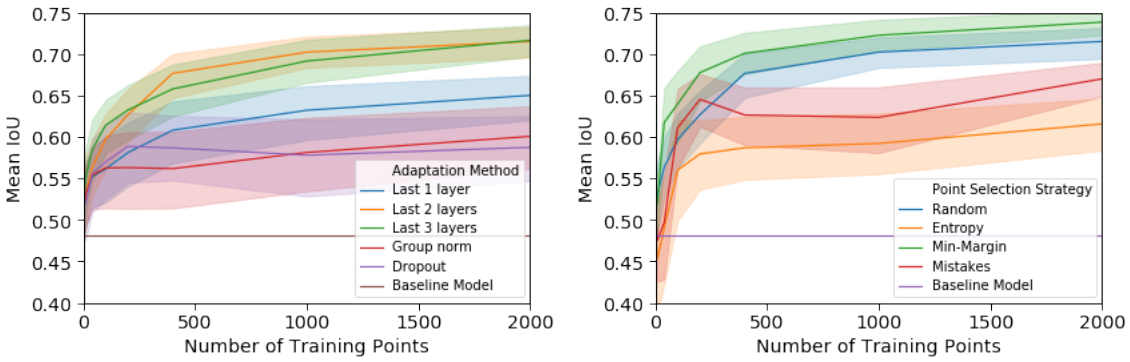


Figure 2.8: Performance of different **fine-tuning** methods (top) and **query methods** (bottom), mean and standard deviation over 5 runs and 4 target areas. At several stages – after 10, 40, 100, 200, 400, 1000, and 2000 points have been seen – the system selects a further set of training points using the given query method and retrains the model using the fine-tuning method. The performance of the model evaluated on the entire target region tends to improve as more points are seen.

2.3.4 Online study of the hybrid labeling system

As can be seen from the indicated confidence intervals in Fig. 2.8, it is not clear that we can expect any of the active learning methods to outperform random selection of data points to label, as was previously often observed in active learning literature [72]. Thus, we test our hybrid labeling system – the HUMAN query method – against the RANDOM point query method. As the LAST 1 LAYER and LAST 2 LAYERS fine-tuning methods tend to perform best in the offline experiments, we also choose them for use in online experiments. [6]

Setup

We recruited 50 users[7] through Amazon Mechanical Turk to implement the HUMAN method using the web interface and interactions described in Sec. 2.3.2. Users use the web tool in a series of 15-minute *tasks*. A *task* is performed in one of four distinct 84km^2 *areas* in New York and using one of two fine-tuning methods chosen above. Before each *task*, the model is reset to the baseline, pretrained only on data from Maryland. Each user performs four *tasks* (one for each *area*, in a random order): in the first three *tasks*, the user uses one type of fine-tuning method, while in the fourth *task* the other fine-tuning method is used. Such an assignment allows us to separate the first task – during which the user is getting used to the tool – from tasks 2 and 3, where the user is assumed to be doing their best work, and from task 4, where the learning system changes its behavior (i.e. where the *fine-tuning method* changes). This allows us not only to measure the variation in performance across users, *fine-tuning method*, and *areas*, but also to see if the users are building an understanding of how the model and its adaptation work.

We use a standard crowdsourcing setup in the Amazon Mechanical Turk system to acquire unbiased ground truth labels on the same four *areas* in New York – we refer to

[6] Precisely, LAST 1 LAYER was full adaptation of the 64×4 parameters in the last (softmax) layer (gradient descent to convergence on all user-supplied points), while LAST 2 LAYERS was a fixed number of iterations of gradient descent on the parameters of the last two layers.

[7] See the supplemental materials for study details.

these labels as the "crowdsourced ground truth labels"[8]. We collected a total of 6009 labels on randomly selected points, from 54 unique labelers, resulting in a dataset of 3441 unambiguous labeled points. These labels agree with the Chesapeake ground truth data 91.1% of the time, which is in line with that data product's published quality estimates [73].

Now, during each *task*, every time the user induces retraining of the model, we calculate that model's performance on the set of crowdsourced ground truth labels from the area in which they are working. We compare this method with the RANDOM query method using the crowdsourced ground truth dataset. In the crowdsourced labeling task, users take ~ 3 seconds to label each pixel they are shown. Thus, in a 15-minute window, they could provide labels on ~ 300 randomly sampled points. A central question is that of *label efficiency*: is human time and money best spent by labeling the central pixels of random patches of aerial imagery (human as *label oracle*) or by using our interactive tool (human as *query method* and *label oracle*)?

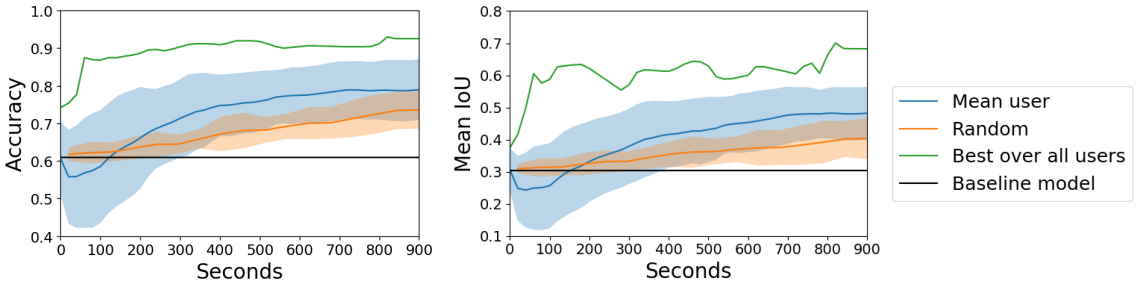


Figure 2.9: Performance of HUMAN and RANDOM query methods for model fine-tuning in a 15-minute time window, measured in pixel accuracy (left) and mean IoU (right). Mean user performance is calculated over the top 50% of users and considers sessions using the LAST 2 LAYER fine-tuning method. Random performance is averaged over 10 seeds, with points assumed to be added every 3 seconds. Both methods are averaged over the same four *target areas*.

Results

The subplots in Figure 2.9 show accuracy and mean intersection-over-union (IoU) of intermediate models achieved at different times in the 15 minute fine-tuning sessions, averaged

[8]See the supplemental materials for study details.

across users. In the case of the RANDOM method, we assume that 300 points are added at uniform time intervals and the model is retrained every 45 seconds. As the model for a specific user will fluctuate in performance over the duration of a single session – users pick up on different deficiencies in the core model at different points during a session – we summarize the HUMAN method as a whole by averaging performance metrics over sets of users. Models fine-tuned using the HUMAN query method consistently outperform models that are fine-tuned with RANDOM queried points, within 3 minutes of labeling (~ 60 samples). The top curve in Fig. 2.9 shows the best model over all users at each point in time, showing that some expert users are in fact able to dramatically outperform RANDOM.

Further analyzing our user results across the four consecutive tasks, we find that users' area-adjusted performance in task 2 is highly predictive of their performance in task 3 ($p < 0.01$, rank-correlation $\rho = 0.4$): of the top 25 (half) of users ranked by (IoU) performance in task 2, 17 are also among the top 25 in task 3. Thus, **the better-performing labelers are detectable** in a statistically significant manner. This indicates that the users are developing different levels of intuition about the inner workings of the network and the fine-tuning method. In addition, the performance of the users in tasks 2 and 3 is far less predictive ($\rho = 0.1$) of their performance in task 4, where the fine-tuning method is switched. This indicates that **users are building a *theory of mind for the AI agent* they work with in tasks 1-3, which is then broken by a slight change in the learning algorithm in task 4**.

An analysis of points submitted by all users in an area show that the users are *not* choosing points to label at random. Different users are drawn to similar parts of the study areas, while other parts remain unlabeled by most users. We explore this further in the "User Attention" section of the supplementary material. Our offline experiments with the MISTAKE method indicate that the model simply knowing where its errors are cannot automatically beat the RANDOM selection of points for labeling. This indicates that **human guidance goes beyond simply quickly spotting errors**, especially for best performers, reminiscent of the super-teacher idea [74]. Text feedback from users (see supplementary material) pro-

vides further interesting insights that should be useful in the design of hybrid systems of this kind.

2.3.5 Discussion

We have conducted a study of hybrid human-AI intelligence on the task of high-resolution land cover mapping. We demonstrate that giving control of the data selection process to the human yields significant improvements in model accuracy. Our user studies show that users develop a theory of mind for the ML system, learning to understand the workings of particular AI algorithms with variation in this skill correlated across different tasks. If we consider this learning framework in the context of usual ML challenges, where humans (engineers/researchers) are paired with machines (new model designs), we can see that similar variation in accuracy comes simply from humans' knack for the particular challenge, even when everyone uses the same architecture and learning algorithms.

By injecting the human into the learning loop, gains from both the human and the AI labor are amplified, not replaced. For the machine, sparse but well-chosen human feedback reduces the cost of computational resources needed to adapt models. For the human, increased sample efficiency of the ML systems acts like an ever-more useful wand with which they can paint the land cover. Together, this collaboration achieves critical cost reduction in practical problems. The Chesapeake dataset was created in 10 months at a cost of $1.3 million, though it covers just 2% of the US [14] with an estimated accuracy of 90%-95%. The best user from our study, averaged over the four target areas, achieved an accuracy of 89.1% in just one hour of labeling work.[9] If such users were to label the entire Chesapeake Bay watershed using our method, this would take 925 hours of work at a labor cost of $18.5k. Of course, other tradeoffs between accuracy and cost are possible by allowing users to work longer on each area or even to work collaboratively.

In problems of massive scale where unlabeled data is practically limitless, such as land

[9]This number is in line with the recent state-of-the-art algorithm [34] which uses 30m low-resolution labels as additional data. Our approach does not rely on the existence of such low-res labels.

cover labeling, it is not likely that a few months of labeling through our tool would create enough training data that the need for human labor would disappear. Instead, applications that are now infeasible, such as quick generalization to new areas or addition of new target classes (shown in supplemental materials), would become feasible, making both the ML algorithms and human labor more valuable than before.

2.4 Self-supervised feature learning

Given an unlabeled dataset $\{x_i\}_{i=1}^n \subset \mathbb{R}^d$, the goal of unsupervised representation learning is to find an embedding function $f(x_i; \phi) : \mathbb{R}^d \to \mathbb{R}^k$ with parameters ϕ that transforms each x_i (typically assumed to consist of "low-level" features, e.g. pixel values of an image), into semantically meaningful "high-level" features $z_i = f(x_i)$ (e.g. what objects are present in an image). After embedding each x_i, the representations z_i can be *transferred* to downstream settings such as arbitrary supervised learning problems or used in data visualizations. An embedding function, $f(\cdot)$, is considered to be "good" if the representations it generates can be effectively used in such settings. Formally, a *transfer task* consists of a relatively small labeled dataset, $\{(x_j, y_j)\}_{j=1}^m$ (compared to the original unlabeled dataset, $m \ll n$), and a loss function encoding the task to be learned from the dataset. An embedding model is effective in this task if we can train a model, $g(z_j; \theta) = y_j$, with parameters θ, using the embedded features to a lower test error than one that uses the original "low-level" features, $g(x_j; \theta)$.

Methods for learning "good" representations from unlabeled data are especially valuable in domains where labels are expensive to collect, however unlabeled samples are plentiful, for example with: electronic health records [75], speech data [76], time-series data [77], and geospatial data [78, 79]. In these settings the learned embedding functions are *necessary* to achieve acceptable performance in the downstream tasks.

In the supervised setting, neural networks will implicitly learn an embedding function as described above. Convolutional neural networks trained on the large ImageNet

dataset, for example, famously learn feature representations that *transfer* well in diverse downstream settings [80]. Here, $f(\cdot)$ is the entire network up to some cut-off layer (e.g. the second to last layer), while $g(\cdot)$ is the remainder of the network. During training, the model must learn an embedding function that creates feature representations which can be used by $g(\cdot)$ (a linear classifier in most cases) to differentiate between the different classes in question. In the unsupervised setting, such a task is not natively possible, as there are no classes to differentiate between. Hence, many unsupervised representation learning methods rely on different forms of *self-supervision* [81], where pseudo labels are created for each input sample, then $f(\cdot)$ is learned as if in a normal supervised setting. Formally, *self-supervised* methods create a pseudo-dataset from the original unlabeled dataset, $\{(l_{in}(x_i), l_{out}(x_i))\}_{i=1}^{n}$, where $l_{in}(\cdot)$ and $l_{out}(\cdot)$ are arbitrary functions that encode the *pretext task* to be solved.

Many existing methods for unsupervised representation learning rely on devising a pretext task based on prior knowledge about the data. For example, [82] observe that "natural images" (i.e. images taken from a camera) are usually oriented such that subject of the image is upright - i.e. have a "photographer bias". They therefore choose $l_{in}(x_i)$ to be a function that applies a rotation of 0, 90, 180, or 270 degrees to an input image x_i, and they choose $l_{out}(x_i)$ to be a function that returns the rotation applied to x_i. The intuition here is that a convolutional neural network trained on this pseudo-dataset must learn features that describe the contents of an image – e.g. the subject, type of the subject, and pose of the subject – in order to determine the "correct" orientation. Other pretext tasks for image datasets include: predicting the relative position of two patches in an image [83], predicting the relative positions of a set of patches (solving jigsaws) [84], and predicting the colors of each pixel from grayscale inputs [85, 86]. Similarly, the temporal information in unlabeled video datasets can be exploited to design pretext tasks such as tracking objects [87] or sorting image sequences in a video [88, 89].

Examining the properties of self-supervised learning, [90] find that self-supervised

training with a combination of pretext tasks[10] in a multi-task learning setup always results in better performance on downstream tasks than training with a single pretext task. Intuitively, different pretext tasks require the network to learn different types of features, some of which do not overlap. Similarly, [92] find that performance on downstream tasks scales with pretext task difficulty[11]. [93] find that self-supervised learning with a single image, using appropriate data augmentation, is sufficient to learn filters that extract low level image statistics in the first layers of CNNs.

The previously mentioned tasks all rely on human-designed functions for $l_{in}(\cdot)$ and $l_{out}(\cdot)$ that encode inductive biases about the structure of the input data and potential downstream tasks. However, these biases can be too specific for certain downstream tasks or input domains. For example, predicting rotation will not be a relevant pretext task with satellite imagery data as there is not a "true" orientation for the network to discover. Similarly, the Exemplar pretext task [91] requires $f(\cdot)$ to be invariant to color, resulting in learned representations that can not take advantage of color.

We expand on this prior work by proposing a simple pretext task: choose $l_{out}(x_i)$ as a randomly initialized CNN. In other words, our proposed pretext task is to learn a network that can mimic the output of an untrained CNN (or, as we experiment with, a set of untrained CNNs). This makes no assumptions about (1) the domain of the input dataset; (2) the structure of the embedding model to be learned; or (3) the representation required by the downstream task. The intuition behind our approach is similar to that of Deep image prior [94] - *randomly initialized CNNs provide a strong prior over natural images*, and can act as an effective substitute for other hand-crafted priors in a variety of inversion tasks (e.g. denoising, super resolution, and inpainting). The output of an untrained CNN will not simply be noise, but will contain useful statistics about the input (see Figure 2.10 for an example). Furthermore, *fitting* a new CNN to this function through stochastic gradient

[10]These tasks include: relative position [83], colorization [85], exemplar [91], and motion segmentation [88]

[11]Specifically, performance increases with the number of permutation options, $|\mathcal{P}|$, that the network must choose between in the Jigsaw task [84].

descent (SGD) based methods will force it to approximate these statistical regularities, and in doing so, will force the $f(\cdot)$ to learn useful features of the inputs.

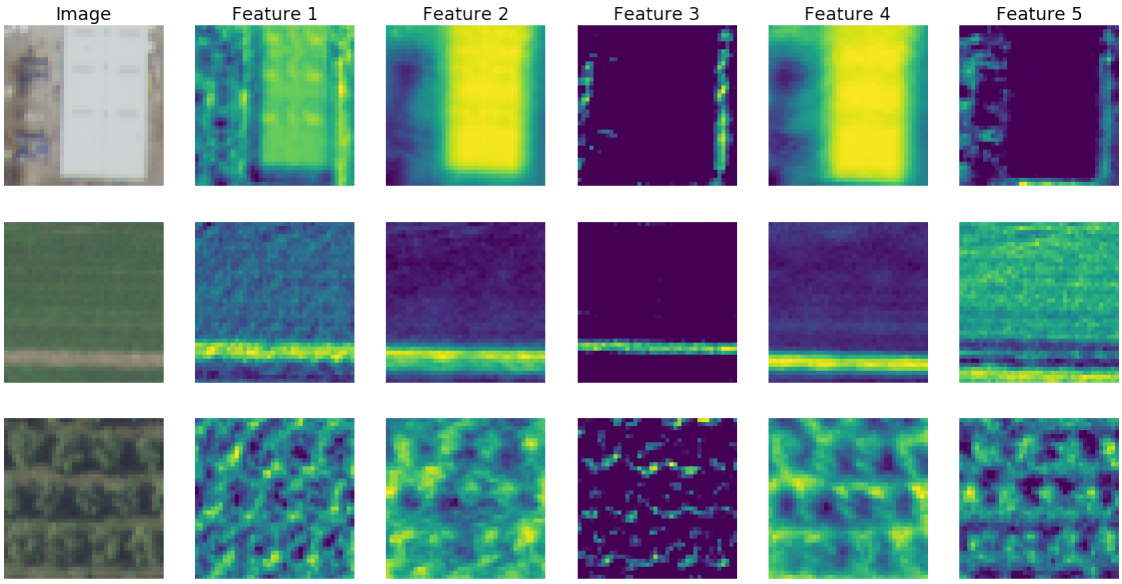


Figure 2.10: (**First column**) $0.6\text{m}^2/\text{px}$ aerial imagery from the National Agricultural Imagery Program (NAIP). (**Other columns**) Selected feature maps from the last layer of a *randomly initialized* CNN (5 convolutional layers weights initialized according to the Glorot uniform scheme [95] and ReLU activations) generated from the imagery. Despite not being tuned in any way, the randomly initialized convolutional filters are able to produce meaningful features due to the structure of the input.

2.4.1 Learning from random functions in the linear case

Formally, the task of learning a feature extractor that can be transfered to arbitrary supervised learning problems can be expressed as:

$$\phi_u = \arg\min_{\phi} E_{h\in\mathcal{H}}[\min_{\theta} E_x[\ell(g(f(x;\phi);\theta), h(x))]. \tag{2.4}$$

where the downstream tasks, h, are sampled from some family $\mathcal{H}$, with a shared loss function ℓ. The learned feature extractor should provide representations that are on average best for all functions in $\mathcal{H}$.

Consider a simple example where the expectation over all target functions and model

fitting are performed in the same space; one might ask if the optimization above can yield desirable results at all with such a free definition of targets. In the literature on domain adaptation, transfer, pretraining, and meta-learning methods, it is usually assumed that the functions of interest share certain similarities, and are thus in need of shared features. On the other hand, we propose to approximately solve (2.4) without making that assumption: We are searching for feature extractors for *arbitrary* functions expressible in a certain form. Are there nontrivial feature extractors like that at all, or may we just as well use a single random function for our feature extractor?

In fact, it can be seen that in case of linear functions f, g, h and a quadratic loss ℓ, the optimal features are meaningful: They capture the principal components of the dataset, x.

Lemma. For linear functions $f(x) = Fx$, $g(z) = b^T z$, so that feature extractor parameters are in $\phi = F$, an $n \times k$ orthonormal projection matrix, and the predictor model is parameterized by $\theta = b$, a $k \times 1$ collapsing vector, and with the target functions of the form $h(x) = r^T x$ (which is the same form as $g \circ f$ with $r = bF^T$), assuming r is distributed as a unit Gaussian $r \sim \mathcal{N}(0, I)$, solving the problem (2.4) is equivalent to searching for the maximally varying data projections Rx,

$$\max_R E_x[x^T R^T R x], \tag{2.5}$$

with $R = F^T F$.

Proof. We first look at the inner optimization over $\theta = b$ for a given function $y = b_g^T F_g x = r^T x$, i.e., the following quadratic optimization:

$$\begin{aligned} m &= \min_b E_x (r^T x - b^T F x)^2 \\ &= r^T (\Sigma - \Sigma F^T F \Sigma^{-1} F^T F \Sigma) r \end{aligned} \tag{2.6}$$

where $\Sigma = E_x(xx^T)$ is the covariance matrix of the input data x.

Next, we compute the expectation over all functions as parameterized by the vector

$r \sim \mathcal{N}(0, I)$,

$$
\begin{aligned}
E_r[m] &= E_r[\mathrm{Tr}((\Sigma - \Sigma F^T \Sigma^{-1} F^T F \Sigma) r r^T)] \\
&= \mathrm{Tr}(\Sigma - \Sigma F^T F \Sigma^{-1} F^T F \Sigma) \\
&= E_x[x^T x] - \mathrm{Tr}(F^T F E_x[x x^T] F^T F) \\
&= E_x[x^T x] - E_x[x^T F^T F F^T F x].
\end{aligned}
\tag{2.7}
$$

Minimizing this over matrices F, and thus solving the problem (2.4) depends only on maximizing the second term, which is the same as the problem of finding directions of maximum variation in the input data (2.5).

This result is a direct consequence of linearity of the model and the use of the square loss. Intuitively, if the data will be queried by a randomly oriented vector r, then the best data compression, in expectation, is the one given by the principal components.

If one of the models f, g or the loss function ℓ is nonlinear, then the principal components (or their rotation) may not provide the optimal features. In fact, it is in general difficult to analyze the problem (2.4). Of course, there is an enormous difficulty in computing expectation over the whole space of nonlinear target functions. In addition, a proper analysis would need to take into account that the practical learning algorithms may not find optimal θ parameters when optimizing for a particular target function nor are they capable of find the optimal universal feature extractor parameters ϕ_u. However the known inductive biases of neural networks, as discussed above, suggests sampling a small number of random neural networks as targets h is sufficient to learn the sort of a feature extractor that would inevitably be learned even if we had enough real labels for the targeted downstream task.

2.4.2 Using the output of randomly initialized neural networks in self-supervised learning

Our proposed self-supervised learning algorithm, RANDOMFUNCTION, involves solving the optimization problem in Equation 2.4 by sampling *target networks*, h, from the family of functions, $\mathcal{H}$ that can be expressed by neural networks (convolutional neural networks in our experiments) with random weights. This family of functions has been shown in prior work [94] to impose a strong inductive bias in vision problems. Given a dataset, we create k pseudo labels for each sample by using k different multiclass *target networks*, $h_k(x)$, with randomly initialized weights. We then use this pseudo (labeled) dataset to fit the parameters of a core feature extractor neural network $f(x)$, using k different classification heads $g_k(z)$, in a normal supervised learning setting. Specifically, each target network will generate a pseudo-label between 1 and C_k, $y_i^k \in \{1, \ldots, C_k\}$, for each datapoint, x_i. Correspondingly, each classification head, $g_k(z)$ will predict a label, $\hat{y}_i^k$, given the features, z_i, generated by our core feature extractor network. We fit the parameters of the feature extractor network and the k classification heads in an end to end fashion using the average cross-entropy loss computed over minibatches of datapoints and their set of pseudo-labels $(X, \{Y_1, \ldots, Y_k\})$.

In practice this algorithm has many free design decisions; in our experiments we focus on the following: the number of target networks, and C_k, the potential number of classes generated by each target network. Our results show that even straight-forward choices for each of these will result in a feature extractor model that has better performance on downstream tasks than simple baselines, however we find further performance improvements when there is diversity between random functions, and in the class outputs of each function. In particular we want to sample (or construct) target functions, such that:

1. There is pairwise "diversity" between the feature representation of the set of target functions we sample. As we want the representations of our core model to be effective for learning a variety of downstream tasks, we would like to learn a set of diverse functions.

2. The pseudo labels created by the target functions are "interesting". First, our target functions should not collapse the entire unlabeled dataset into a single class or feature (as this does not provide any signal). Second, some target functions should capture low frequency patterns in the data, while others should capture high frequency patterns.

We propose two algorithms in the following sections for sampling the target functions that encourage these two properties.

Sampling for inter-function diversity

We observe that although random sampled neural networks will extract meaningful features (e.g. in Figure 2.10), that two randomly sampled networks create feature representations that are near linear transformations of each other. Formally, we decompose a target function into a *feature extractor* and *classifier* as $h_k(x) = h_k^C(h_k^F(x))$ (i.e. similar to the way we represent the base network). We measure the similarity between a pair of randomly sampled target function feature extractors, $h_i^F(x)$ and $h_j^F(x)$ over a dataset X as the variance weighted average R^2 value of a multi-linear regression model from $h_i^F(X)$ to $h_j^F(X)$. For the datasets and networks we test with we find that the average feature similarity values for a randomly sampled pair of networks is high. For example, Figure 2.11 shows the distribution of this similarity for 1000 randomly sampled networks over the CIFAR-10 training set - with a mean of 0.68.

Given this, we propose the following algorithm in order to find a pool of target functions with diverse feature representations given an unlabeled dataset: (1) sample an initial random network and add it to the pool; (2) sample a candidate network and measure its average similarity with all networks in the pool; (3) if this similarity is less than a threshold value then add it to the pool, else discard it; (4) repeat from step (2) until the desired number of networks is found. For a given dataset we set the threshold value as the R2 value representing 10% of the probability mass of the distribution shown in Figure 2.11.

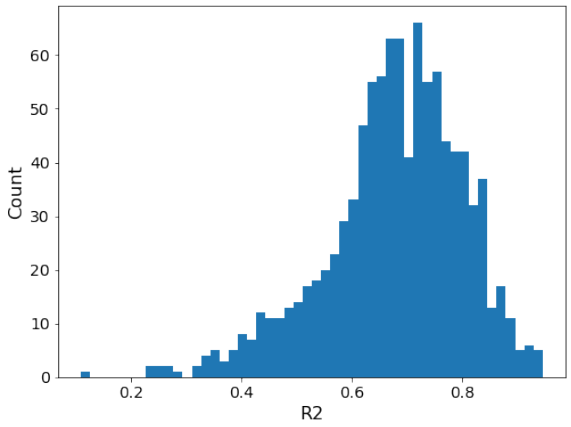


Figure 2.11: Distribution of feature similarity values for randomly sampled CNNs (ResNet18s and 5-layer FCNs) over the CIFAR10 training set.

Optimizing for class diversity in a single network

We also observe that target functions can extract meaningful features through $h^F(x)$, but also destroy that representation through the target function *classifier*, $h^C(h^F(x))$, as the classifier also uses randomly sampled weights. In all of our experiments we consider linear target function classifiers that will multiply a representation z_i by a weight matrix, $W \in \mathbb{R}^{|z_i| \times C}$, in order to generate the pseudo label (in the range $\{1, \ldots, C\}$) for a data point, $h^C(z_i) = \text{Softmax}(z_i^T W) = y_i$. In an extreme case, this may result in every data point being assigned to the same pseudo label, thereby removing all information from the target function. Empirically, we observe that, regardless of C, an entire dataset will be partitioned into few unique classes by a randomly initialized network (Figure 2.12). To avoid these situations, we propose the following optimization step to initialize the weights of the last layer of a target function given an unlabeled dataset:

$$W^* = \arg\min_W \quad \alpha \left[\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} -Y_{ij} \log Y_{ij} \right] - \beta \left[\sum_{j=1}^{C} -Y_j^{\Sigma} \log Y_j^{\Sigma} \right] \tag{2.8}$$

where

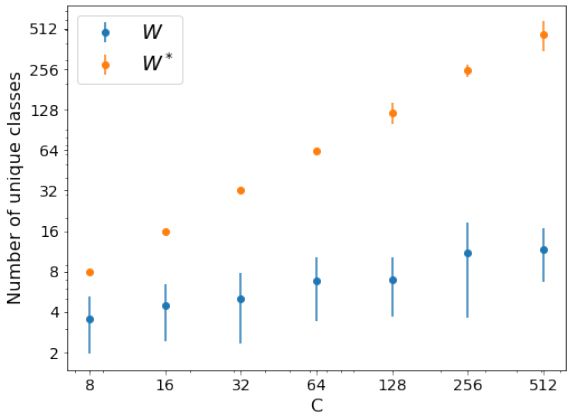


Figure 2.12: Observed number of classes vs. dimension of the target function classifier for CIFAR10 under randomly initialized weights, W, and weights optimized for class diversity, W^*.

$$Y_j^{\Sigma} = \frac{\sum_{i=1}^{N} Y_{ij}}{\sum_{i=1}^{N}\sum_{k=1}^{C} Y_{ik}} \quad \forall j \in \{1, \ldots, C\}$$

Here we want to minimize the average entropy of the predicted class distribution over all samples in the dataset while maximizing the entropy of the class count distribution, Y^{Σ}. In other words, we want the target function *classifier* to "confidently" assign each sample to a unique class, and assign approximately the same number of samples to each potential class. This objective function is fully differentiable with respect to W, and we can solve it using gradient descent. We start the gradient descent process with $\alpha = \beta$ and increase the weight of β every time that the objective value plateaus.

2.4.3 Experiments

Our experiments consist of a **self-supervised learning phase**, where the parameters of a CNN $g(f(x;\phi);\theta)$ are fit according to a self-supervised learning method, then a **supervised learning phase**, where the parameters of f, the embedding function, are frozen and a new function $g'(f(x;\phi);\theta')$ is fit on a relatively small labeled dataset.

We use all available input data (i.e. ignoring the labels) from a dataset in the self-supervised learning phase, then, in the supervised learning phase, we subsample a *fine-tune*

training set from the dataset's training set to fit the parameters of $g'(\cdot)$ and test on the dataset's testing set. We vary the size of the *fine-tune training* set to test how the features learned in the self-supervised phase are able to generalize given different amounts of data to fine-tune with.

Further, we restrict the architecture of $g'(\cdot;\theta')$ to a single linear layer followed by a softmax activation (i.e. a logistic regression model) to test the quality of the representation learned by $f(\cdot)$, and not, e.g. the ability of $g'(\cdot)$ to capture non-linear relationships between the representation and target task[12].

We compare our self-supervised method, RANDOMFUNCTION, with other self-supervision methods: TILE2VEC [78], ROTATION [82], DEEPCLUSTER [96], and Contrastive Predictive Coding (CPC) [97], as well as three simple baselines: RANDOMLABELS, RANDOMFEATURES, and COLOR. RANDOMLABELS is a method that assigns a pseudo label between 1 and C_k at random to each sample. RANDOMFEATURES is a method that skips the self-supervised learning phase and randomly initializes the parameters of $f(\cdot)$. Finally, COLOR is a fixed feature extractor $f(\cdot)$ that computes the channel-wise mean, standard deviation, min, and max statistics over all spatial locations in an input image and returns these statistics as a vector to be used as features in the supervised learning phase.

Datasets, models, and training details

We experiment with a standard vision dataset, **CIFAR10** and two geospatial datasets, **Cropland**[13] and **Chesapeake** [98].

CIFAR10 This dataset consists of 32×32 pixel RGB images from 10 image categories (airplane, bird, cat, etc.). We use the standard 50,000 image training and 10,000 image testing splits. We train all methods using a ResNet18 architecture that uses 3x3 convolutions with stride 1 in the first layer, skips the first max pooling layer, and only has a single block in the fourth group of blocks.

[12]This is known as a linear probe in unsupervised representation learning literature [93].
[13]Reproduced from Jean et al. [78].

Cropland We duplicate the dataset described in [78]. This consists of National Agricultural Imagery Program (NAIP) 2016 imagery (from the California dataset) and Cropland Data Layer (CDL) 2016 labels for the area "spanning latitudes [36.45, 37.05] and longitudes [-120.25, -119.65]". Here, images are 50×50 (pixel) patches of NAIP imagery (covering $30m^2$ areas) and are paired with a single label of the most frequently occurring CDL class in the patch. We randomly sample 110,000 of such patches to use as the training set, and 10,000 labeled patches to use as the testing set (we sample training and testing patches from larger non-overlapping tiles to prevent label leakage due to proximity). We train all methods using a ResNet18 that uses 3x3 convolutions with stride 1 in the first layer and skips the first max pooling layer.

Chesapeake We use the NAIP 2013 imagery and National Land Cover Database (NLCD) 2012 labels from the *Maryland* split in the land cover dataset described in [98, 2]. Here, images are 256×256 crops of NAIP imagery (covering $256m^2$ areas) and are paired with the single label of the most frequently occurring NLCD class in the central 30×30 pixels. We use the 50,000 sampled patches from the *training set* as the training set and the 2,500 labeled sampled patches from the *validation set* as the testing set. For the TILE2VEC method only we sample 50,000 triplets from the tiles in the *training set*. We train all methods using a ResNet18 that uses 3x3 convolutions with stride 1 in the first layer.

Given a dataset we test all methods using a variant of the ResNet18 architecture. In the CIFAR10, Cropland, and Chesapeake datasets we replace the first convolutional layer in the ResNet with one that uses 3x3 convolutions (instead of 7x7 convolutions) with stride 1 in order to learn smaller features. In the CIFAR10 and Cropland datasets we additionally skip the first max pooling layer due to the small input image sizes.

2.4.4 Results

Varrying size of fine-tuning dataset

Figure 2.13 shows final test performance of the self-supervised methods across the CIFAR10, Cropland, and Chesapeake datasets with varying amounts of training data used in the supervised learning phase. Results are reported as an average of 5 runs across both the self-supervised and supervised learning phases using different seeds. The RANDOMFUNCTIONS (DIVERSE) results use 8 random function with $C = 500$ and employ the sampling and diversification strategies described in the previous sections. ROTATION uses rotations of 0, 90, 180, and 270 as described in [82] and DEEPCLUSTER uses $k = 100$.

We observe that RANDOMFUNCTIONS (DIVERSE) method achieves the best or is tied with another method for best performance at all sample levels in these three datasets, except for in the 5% and 10% cases in the Cropland dataset, and the 0.1% case in the Chesapeake dataset. In the CIFAR10 dataset, the second best method is ROTATION, however in the geospatial datasets ROTATION achieves similar performance to RANDOMLABELS, as the rotation pretext task is not relevant to geospatial imagery (there isn't a "photographer" bias in the input images that can be learned from). Similarly, the TILE2VEC method performs well in both geospatial datasets, however isn't applicable to the CIFAR10 dataset, or indeed any other dataset that doesn't have a natural measure of geographic distance between samples. Finally, COLOR performs suprisingly well in both geospatial datasets and is the second best method in the Chesapeake dataset over most numbers of sample points.

2.4.5 Effects of hyperparameters on learning from random functions

We test the effect that the hyperparameters k, the number of random functions that are used in the self-supervised learning phase, and C_k, the potential number of classes that each of the random function can partitions the data into, have on the downstream performance of a self-supervised model tuned on the CIFAR10 dataset. Figure 2.14 shows that downstream

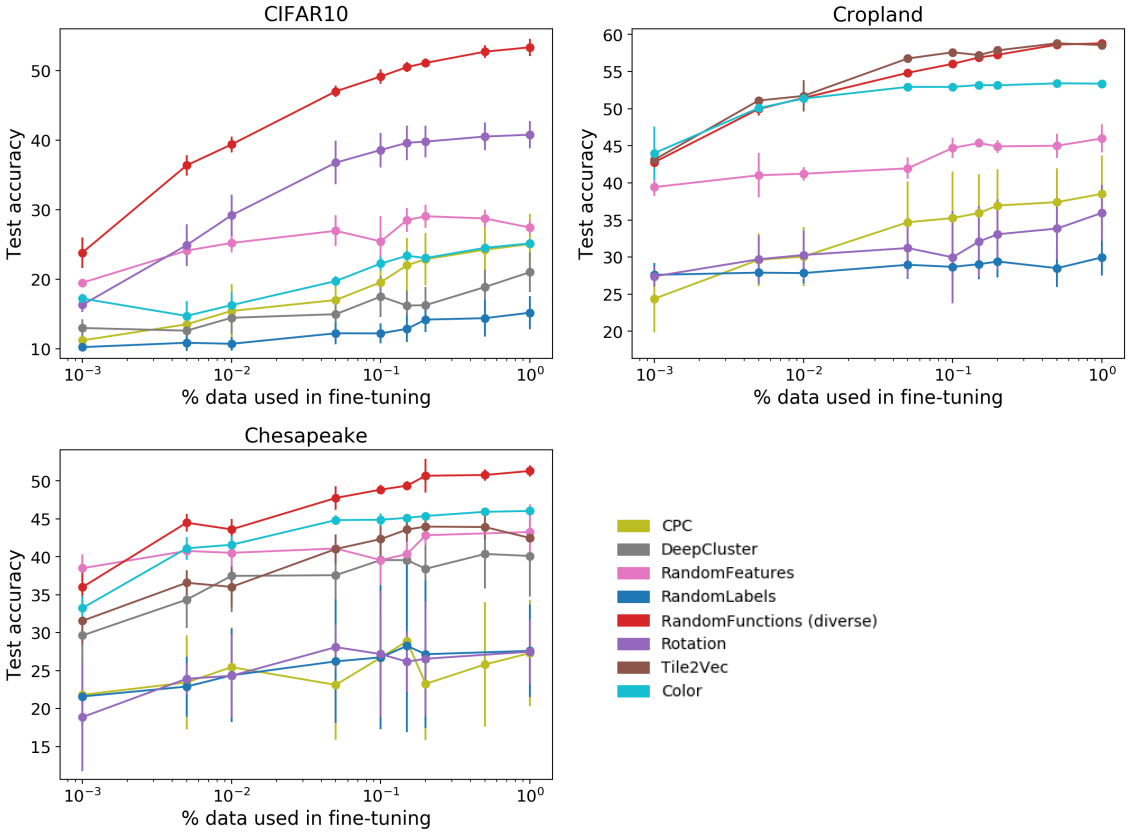


Figure 2.13: Comparison of test accuracy on the downstream task for all self-supervision methods using varying amount of data to fine-tune with.

test performance of RANDOMFUNCTIONS generally increases with number of classes, C_k, and to a lesser extent the number of random functions.

Filter visualization

In Figure 2.15 we show 11x11x3 convolutional filters learned in the first layer of a ResNet18 (modified to use 11x11 filters instead of 7x7 filters) on the **Chesapeake** dataset from fully supervised training on the real labels, self-supervised training with RANDOMFUNCTIONS, and self-supervised training with RANDOMLABELS. We observe that the RANDOMFUNCTION training is able to learn filters with structures that are more similar to those learned with real labels, than with random labels, despite the fact that the random functions themsevles contain random filters (similar to the "Initial filters" shown in the figure). In contrast, the filters learned with RANDOMLABELS simply pick up on different frequent colors in the

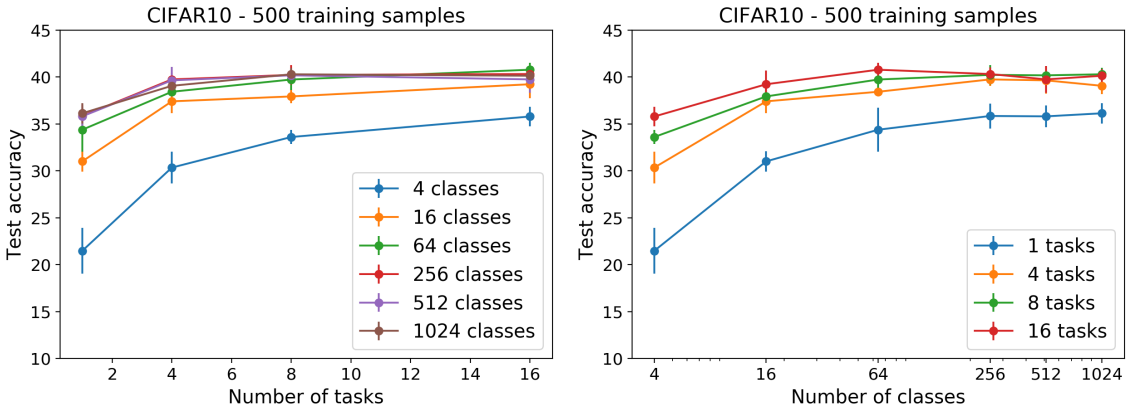


Figure 2.14: Effect of the number of random functions and number of classes that each random function can assign a sample to on the downstream performance in the CIFAR10 dataset.

dataset as there is no signal (by design) between the inputs and labels.

2.4.6 Discussion

In this chapter we have described a method for using randomly intialized CNNs to generate pseudo-labels as a self-supervised learning pretext task. Our proposed method ties and outperforms the state of the art Tile2Vec [78] method for representation learning with satellite imagery in two satellite image classification datasets. While not tailored to satellite iamgery, our proposed method is an important addition to the landscape of methods for learning with remotely sensed data. In particular, self-supervised methods are important in learning tasks with remotely sensed data due the widespread availability of such data but paucity of labeled data to support training highly parameterized deep learning models.

2.5 Human population density estimation

Given an administrative area, the spatial distribution of the population in that area can be determined by answering two questions: “how many people live in the area?”, and “where, specifically, in the area do people live?”. These two questions can be cast as the following two problems: population projection, and population disaggregation. Traditionally, these

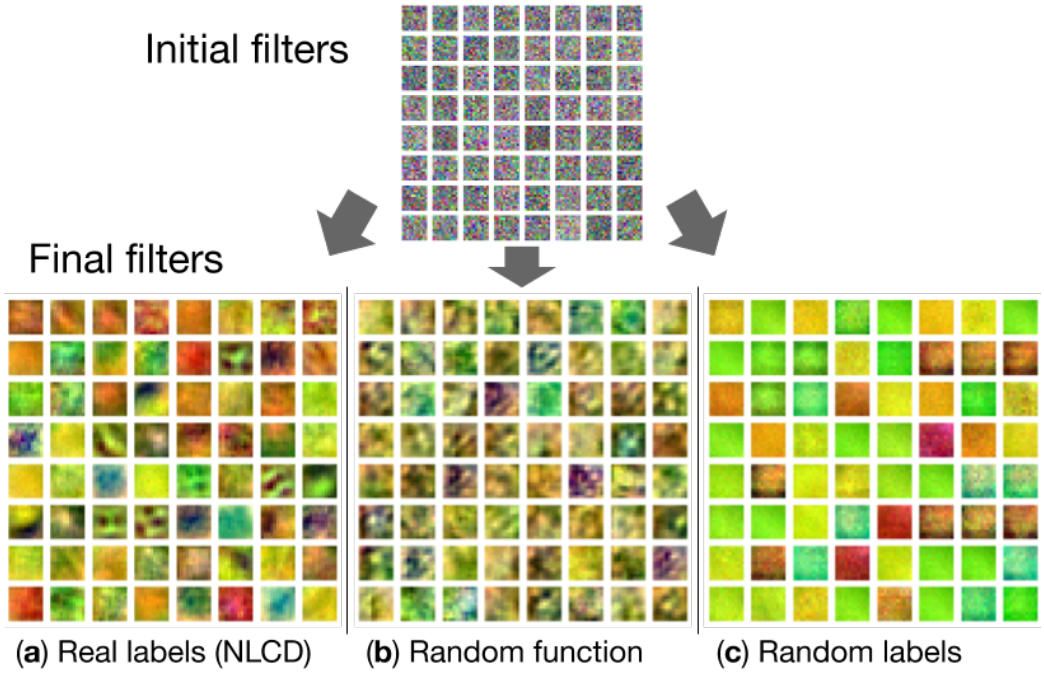


Figure 2.15: Visualization of 11x11 convolution filters in the first layer of a ResNet learned with the **Chesapeake** dataset. (a) shows the filters learned in a fully supervised setting, where the network must classify the image according to a ground truth land cover label. (b) shows the filters learned with RANDOMFUNCTIONS, i.e. from predicting pseudo labels generated by a randomly initialized CNN. (c) shows the filters learned under the RANDOMLABELS baseline self-supervised task, i.e. from predicting a random labels assigned per image.

questions are addressed independently of one another using population projection methods and population disaggregation methods, respectively. In the population projection task, the goal is to estimate the number of people that live in a particular administrative area based on historical data. Methods such as regression models, and non-comprehensive supplemental census surveys (like the American Community Survey) belong to this category. In the population disaggregation task, the goal is to distribute a population estimate for a given administrative area within that area, i.e., at a higher spatial resolution than the population estimate was originally made for.

Our proposed method performs both of these tasks jointly. Using recent techniques from deep learning, which has shown remarkable state-of-the-art results in many computer vision tasks [99, 100], we train convolutional neural networks (CNNs) to directly predict

the population of a given $0.01^\circ \times 0.01^\circ$ area using only satellite imagery, then summarize the predictions at different administrative area resolutions. These high-level predictions provide greater confidence in the accuracy of our model's predictions at the finer resolution. We perform two types of model validation. Quantitatively, we compare our model's grid cell estimates aggregated at a county level to several US Census county level population projections. Qualitatively, we directly interpret the model's predictions in terms of the satellite image inputs.

2.5.1 Related work

Deep learning is being used with increasing frequency to solve problems in the domain of computational sustainability and urban planning. At a broader level, CNNs have been extensively used in computer vision applications in recent years, and have achieved state of the art results in image classification and object recognition [100, 99, 101]. New types of network layers, such as batch normalization and dropout, have also been developed to improve the accuracy of CNNs [102, 103]. Convolutional neural networks have been used to predict the spatial distribution of poverty in developing countries by using nighttime lights as a data rich target for a transfer learning task [104, 105]. Pre-trained CNNs have recently been shown to be effective at the problem of remote sensing image scenes classification through the tuning a small number of layers [106, 107]. Similarly, deep learning has been shown to be effective in the task of classifying land cover type, with recent work that has achieved high classification accuracy on new large land cover datasets using mixed CNN based approaches [108, 109].

The most similar work to ours also uses CNNs to estimate population from satellite imagery [110]. The motivation of this paper is similar to ours, as we both attempt to create high-resolution gridded population counts for use in planning applications. This paper estimates population in Kenya at a 8km^2 resolution with a CNN trained on data from Tanzania at a $250m^2$ satellite pixel resolution. The author's propose a way to use

their CNN's output as a weighted surface for population disaggregation, and compare this method to other methods for disaggregating population counts in Kenya. Our work differs in several important ways. First, we focus on validating our model's predictions as raw population projections and do not consider using our model's prediction as a weighted surface for distributing population counts. If the population (or projected population) of an area is known a priori, then any population *assignment* method can degrade into a weighting scheme. Secondly, we focus on interpreting the results of our model as a way of validating its ability to generalize. Thirdly, we apply our method to the entire US using census block derived training and testing data.

Other related work is divided between the two problems we aim to address jointly with our method: population projection and population disaggregation. In the following paragraphs we address each of these problems to give context to our methodology.

On average, county population can be reliably extrapolated over short time horizons with simple linear models, however if some counties experience disproportionally higher or lower growth rates, more complicated models are needed [111]. The US Census has led research into population and demographic projections, and uses a variety of different population and demographic projection methods to create sub-national projections broken down by age, sex, and race [112, 17]. Census postcensal projections, projections done in between census years, are created with a method known as the ratio-correlation method [113, 114, 17]. This method uses the current year's estimated population, number of live births, registered vehicles, public school enrollment, registered voters, deaths, and other information to determine the estimated population change at the next census date. More recently, the American Community Survey has been used as annual supplemental surveys to update the demographics profiles of a variety of sub-national areas in between census years [115, 116].

Population disaggregation methods, and the creation of high resolution population grids have been studied for decades [117, 118]. The most basic method in this class is areal inter-

polation, whereby the known population of an administrative zone is distributed uniformly across its area [119]. This process happens on a discretized grid over an administrative zone, where each cell in the grid is assigned a population value equal to the total population over the total number of cells that cover an administrative zone. Dasymetric weighting schemes extend this idea of distributing the known population of an area by creating a weighted surface to distribute the known population, instead of doing so uniformly. The weighting schemes are determined by combining different spatial layers (e.g., slope, average rainfall, land/water masks) according to some set of rules. While some weighting schemes are completely ad-hoc, recently, machine learning methods have been used to improve upon this approach [120, 121, 122]. These methodologies are similar to traditional supervised machine learning problems [123], but since actual ground truth data does not exist to compare against, validating the results of dasymetric models is challenging. Finally, there are many existing gridded population datasets created using a variety of the previously mentioned disaggregation techniques. Briefly, these include: Gridded Population of the World [124], GRUMP [125], Landscan [126, 127], as well as the AfriPop, AsiaPop, and AmeriPop databases.

2.5.2 Methods

The goal of this research is to make high-resolution gridded population estimates from satellite imagery. To do this we train CNNs that take satellite imagery of some area as input, and output a population estimate for that area. We train our models on the continental United States using US Census population counts and Landsat 7 1-year composite imagery from the year 2000. We test our models using the 2010 versions of the same datasets, and evaluate the population estimates in two ways: (1) aggregating our model's estimates at the county geography level, then comparing them to projected county population counts; and (2) showing *why* our model makes predictions in terms of input image features.

As described in Section 2.5.2, we let $\mathbf{P}_t$ be a grid of target population values covering

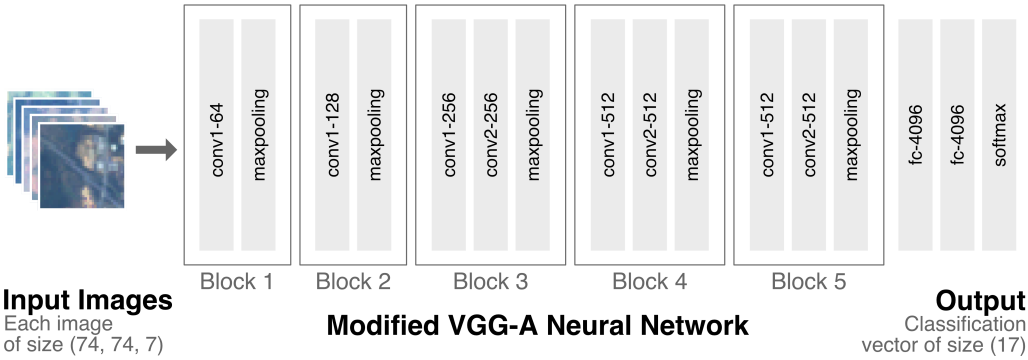


Figure 2.16: Our deep learning model architecture, based off of the VGG-A model. The model inputs satellite images of size (74, 74, 7) in to a linear neural network consisting of 5 convolutional blocks. Each convolutional block contains at least one convolutional layer (conv) and a maxpooling layer. After the 5 convolutional blocks, two fully connected (fc) layers feed into the softmax activated output of length (17) to perform classification.

the continental United States, $\mathbf{C}_t$ be a grid of target population class values, and $\boldsymbol{\theta}_t$ be a grid of satellite images, where for every target value $P_t^{i,j}$ and $C_t^{i,j}$ there is an associated satellite image, $\theta_t^{i,j}$. Using this notation, we can express our learning task as estimating two functions: one in a regression format, $f(\theta_t^{i,j}) = P_t^{i,j}$, and one in a classification format, $g(\theta_t^{i,j}) = C_t^{i,j}$. For the purpose of this study we will focus on the classification version of this problem. We use CNNs to approximate this function, as the mapping from image to population counts will be highly non-linear, noisy, and depend strongly on the semantic content of the input image, e.g., on the quantity and type of buildings visible in an input image. Once we have approximated g on a training year, i.e. for $t = 2000$, we can use it to create population projections for a future year, in which a census has not been taken, but satellite imagery exists for. We validate this modeling methodology by training CNNs using data from $\mathbf{C}_{2000}$ and $\boldsymbol{\theta}_{2000}$, then running our model with all of $\boldsymbol{\theta}_{2010}$ to create a predicted population surface for 2010. To evaluate our predictions, we compare our predicted population values aggregated at the county level to other county level population predictions, we show the errors our models makes, and we use interpretation techniques to uncover why our models are making such predictions.

We describe the data and the preprocessing steps that we use in Section 2.5.2, the CNN

model architecture choices in Section 2.5.2, and the experimental methodology that we follow to train, validate, and test our models in Section 2.5.2. Note that we perform all model training, testing, and experiments using a single desktop workstation containing an NVIDIA Titan GPU.

Data

We use three datasets in this work: the Center for International Earth Science Information Networks' (CIESIN) US Census Summary Grids for 2000 and 2010 [128, 129], Landsat 7 1-year composite images for 2000 and 2010 (courtesy of the U.S. Geological Survey)[14] downloaded from Google Earth Engine, and county level population data for 2000 and 2010 from the US Census.

The US Census Summary Grids are raster files with a resolution of 30 arc-seconds ($\approx 1km^2$) where the raster cell values are population counts from their respective census. The per cell counts are created by disaggregating census survey data from census block geographies, while taking into account various geographic features, such as bodies of water, where people won't be living. In general, a raster cell will contain an area-weighted combination of the populations from the census block shapes that it intersects with. Since census block geographies are smaller than the 30 arc-second grid in heavily populated areas, these maps represent the closest "ground truth" values for population that are available to use as training data for our machine learning models. As a pre-processing step, we re-project these two rasters into a slightly coarser grid with a resolution of $0.01^\circ \times 0.01^\circ$ ($\approx 1105m^2$ at the equator), where the northwest corner is at $124.849^\circ W, 49.3844^\circ N$.

We represent each of these grids as a matrix, $\mathbf{P}_t \in \mathbb{Z}_+^{2499\times5796}$, where an entry $P_t^{i,j}$ represents the population of the cell in the i^{th} row and j^{th} column from year t (in this case $t \in \{2000, 2010\}$). We further pre-process the data by creating an additional, binned version of each population raster, where a cell takes on a value representing which bin its

[14]Landsat: `https://landsat.usgs.gov/`

population count falls in. Specifically, we create matrices $\mathbf{C}_t$, where an entry $C_t^{i,j} = 0$ if $0 \leq P_t^{i,j} < 1$, 1 if $2^1 \leq P_t^{i,j} < 2^2$, ..., k if $2^k \leq P_t^{i,j} < 2^{k+1}$ where $k \in \mathbb{N}$. This process discretizes the target population values which simplifies our learning tasks by creating a classification problem. For $\mathbf{C}_{2000}$ the highest class value is $k = 17$, representing a cell that has a population in the range $[65,536,131,072)$. For the rest of the study, we will use these *population class values* instead of the raw population count values when discussing estimating population.

Landsat 7 1-year composite data is available through Google Earth Engine for the years of 1999 through 2014[15]. The 1-year composites are made by taking the median pixel values from a sample of the least cloudy images from the given year. We use data from the 2000 and 2010 sets, with bands 1 through 7, at a $15m^2$ resolution. This data is downsampled from the native resolution of $30m^2$ recorded by the Landsat 7 satellite using nearest neighbor interpolation. As a pre-processing step, for every $0.01^\circ \times 0.01^\circ$ cell in the population matrices, we take the grid of Landsat imagery that it covers. We resize the grid of Landsat imagery covered by a single population cell into a square volume with a height and width of 74 pixels, as the number of actual satellite imagery pixels that cover a $0.01^\circ \times 0.01^\circ$ area will vary with latitude. We choose a height and width of 74, because at a latitude of $45^\circ N$ (approximately the center of the US), a $0.01^\circ \times 0.01^\circ$ cell is $\approx 1,111m^2$, and with a height and width of 74 pixels of 15x15 meters, our satellite images will represent a similarly sized $1,110m^2$ area. We let the grids of Landsat images be represented as $\boldsymbol{\theta}_t$, where by for every $P_t^{i,j}$ cell from the population matrices, we have an associated satellite image volume, $\theta_t^{i,j} \in \mathbb{Z}_+^{74\times74\times7}$.

The county level population data from the US Census includes the ground truth population values for each county in 2000, and 2010, the postcensal population estimates for each county in 2010, and the ACS 5-year 2006-2010 population estimates for each county in 2010. We use this data evaluate our models' aggregate estimates, and refer to the ground

[15]Google Earth Engine: `https://earthengine.google.com/`

truth 2010 county population counts as "Actual 2010" in Section 2.5.3.

Model Architecture

We experimented with different CNN architectures and hyperparameters using training and validation sets sampled from the 2000 datasets over a $1^\circ \times 1^\circ$ area in the southeast United States. Our assumption is that a model architecture/hyperparameter set which can perform well on this subset of the entire US will be able to perform equally well throughout the entire study area. The training and validation set sampling was performed through the methodology described in Section 2.5.2.

We considered the 5 well-known 'VGG' model architectures, VGG-A through VGG-E from [99], and variations of each of the 5 VGG architectures that included dropout and batch normalization layers. We adapt the VGG architectures to use our input images of size (74,74,7). Since we have discretized our target values into 17 different classes, we resize the output layer to 17 and use a softmax activation function. For all experiments we use a batch size of 512 samples, the Adam optimization method [130] from the Python Keras library [131] (with default parameters), the categorical cross entropy loss function, and we train all networks for 30 epochs (with consideration to overfitting through observing the training/validation loss curves). We found that a VGG-A architecture results in the best top-1 and top-3 accuracy on both the training and validation sets over 30 training epochs and therefore use this architecture for the remainder of the study. See Figure 2.16 for a diagram showing the structure of our model. We chose 30 epochs as a cut off as the best models do not show any improvements in terms of validation loss after this point.

Experimental Setup

Our study area consists of a $2,499$ by $5,796$ grid covering the continental United States that contains ≈ 8 million target values. As using all of these samples to train with presents a significant computational challenge, we divide up the study area into 15, $1,000$ by $1,000$

($1^\circ \times 1^\circ$) *chunks*, and train an independent model for each chunk according to the methods described in Section 2.5.2. Recent work using random forest models for population mapping suggests that, "more accurate population maps can be produced by using regionally-parameterized models where more spatially refined data exists" [120], which we follow with this methodology. Within each chunk we sample $1/10$th of the available data to use as training samples, and $1/100th$ of the data to use as validation samples. As there is a class imbalance problem in the population data, with many more samples in the lower population classes than in the higher population classes, we perform a weighted sampling to select training and validation points. We let c_i represent the number of points in class i over the entire training set, then the probability of selecting a point $C_t^{i,j} = x$ is given as $1 - c_x / \sum_{i=1}^{17} c_i$. This sampling methodology serves to undersample the higher frequency classes more often than the lower frequency ones, while still resulting in a representative sample of all classes from the study area. Figure 2.17 shows the results of this sampling methodology.

An important component of any machine learning or modeling application is validating that the models are able to generalize well to unseen data, and that the models are able to make reasonable predictions. It is important to note that because there does not exist any true "ground truth" gridded population data, it is not possible to truly evaluate population disaggregation techniques. As the purpose of our models is to predict population values from only satellite imagery, they should (a) be able to make reasonable population predictions when compared to other population prediction techniques, (b) be interpretable, where population predictions are able to be explained in terms of semantic features of the input images, and (c) should have explainable errors. We address each of these three points in the following three paragraphs.

We first evaluate our results by comparing our model's aggregate population estimates at the county level with US Census Postcensal county level estimates for 2010 (**POSTCENSAL**) [17], and American Community Survey 5-year estimates for 2006-2010 (**ACS5YR**) [116]

in terms of accuracy when evaluated against the actual 2010 Census [129]. We convert our per grid cell population class predictions, $\hat{C}^{i,j}$, into county level population estimates, $\hat{P}^{i,j}$, in two ways. The first method (**CONVRAW**), involves converting the class values directly into population values as described in Equation 2.9.

$$\hat{P}^{i,j} = \begin{cases} 0 & \hat{C}^{i,j} = 0 \\ \frac{1}{2}(2^{\hat{C}^{i,j}-1} + 2^{\hat{C}^{i,j}}) & \text{otherwise} \end{cases} \tag{2.9}$$

This formula is equivalent to predicting the middle point of each class bin as the population estimate. We sum the predicted population values for each cell whose centroid falls within a particular county to get the aggregate county predictions. The second method, (**CONVAUG**), involves using the values from the softmax activations in the last layer of each CNN as "features" into a secondary machine learning model. Specifically, the last layer of our CNN models has a width of 17, where the output values represent the probability that the input image belongs to each of the 17 population classes. We run our CNN models for each cell in the training dataset (covering the entire US), and record the output vector at each location. We aggregate the output vectors by county by summing the vectors of all pixels that are covered by each county. This process gives us a *feature vector* for each county which contains information about the composition of the population classes of the cells that make up that county. We then use these feature vectors to train a gradient boosting model to predict the ground truth county population values from the training set year. We perform the same process on the test set to create feature vectors with our trained CNN models and use the trained gradient boosting model to make county level population estimates. While this methodology is somewhat orthogonal to the main points of this paper, it shows how our trained CNN models can be used as a mechanism for feature extraction, and that the features the model learns are indeed valid signals of population numbers. We show the results from this county level evaluation in Section 2.5.3.

As described in the previous paragraph, for each input cell our model outputs a prob-

ability distribution over the possible population class values. Using this, we create maps that show the probability that each cell belongs to a given class. Similarly, we show which input images maximally activate every given output class. We show these interpretability results in Section 2.5.3

Finally, we interpret the largest errors that our model makes. Because our model is limited to using satellite imagery data, it will become "confused" in cases where there are signs of human settlements that do not manifest as populated in the census datasets. This confusion is evidence that our models are able to learn the higher-order features as to what constitutes "populated areas", however do not have enough data to discriminate between different types of human activities. The results and discussion of this are shown in Section 2.5.3.

2.5.3 Results

Our results focus on validating the modeling methodology, and are broken down into three sections: evaluating how good our model's population estimates are when aggregated at the county level in Section 2.5.3, interpreting why our models make the predictions that they do in 2.5.3, and evaluating and explaining our model's per pixel errors when compared with ground truth in Section 2.5.3.

County level Estimates

Here we compare 4 different methods for predicting county level population counts for the continental US in 2010. The four methods are as described in Section 2.5.2: **POSTCENSAL**, **ACS5YR**, **CONVRAW**, and **CONVAUG**. None of these methods contain information about the true population counts for the target year, 2010, therefore must infer the population either from detailed historical population and demographic data in the case of **POSTCENSAL**, supplemental survey information in the case of **ACS5YEAR**, or a combination of satellite and historical population data in the case of our methods **CONVRAW**

and **CONVAUG**. We compare the predicted populations for all counties with each method to the ground truth population taken from the US 2010 Census and record the mean absolute error (Mean AE), median absolute error (Median AE), r^2 score, and mean absolute percentage error (MAPE). The results for this comparison can be found in Table 2.4, and the per county errors for each method are visualized in Figure 2.18.

The two statistical methods used by the US Census provide more accurate predictions of county level population for 2010, and have lower median and mean absolute errors than our two methods. This result is expected, as the predictions made by these methods take many more historical features into account, while our methods only use the previous census' population counts and satellite imagery to make predictions. Our model's mean and median errors fall within an order of magnitude of the census model's errors, and our model's MAPE is similar to the ACS5YR results. We perform this comparison to validate that our model's unaided population estimates are not wildly off, which suggests that our model is able to capture the true signal in determining population values from satellite imagery. Considering the evaluation of how well our model captures the *locations* of populations, we argue that because our aggregate estimates at the county level are not wildly off, our model's individual cell predictions must be approximately valid as well. Similar to population disaggregation methodology, our model's individual cell predictions will be the most accurate when they are scaled to match the true population value, or a trusted population estimate. While these county level estimates should not be used in place of the more accurate census estimation methods in the US, they could be used to create continuously updated population maps for developing countries that do not have the detailed data required to run population projection models.

Prediction Interpretability

Interpretability is an important aspect of any modeling process. Some population disaggregation methods rely on ad-hoc rules to assign the population of an administrative area to the

Table 2.4: County level population projection results. Comparison of 4 techniques for estimating 2010 county population for all counties in the continental United States.

	Mean AE	Median AE	r^2	MAPE
CONVRAW	23,005	6,357	0.9103	73.78
CONVAUG	19,484	4,642	0.9365	49.82
POSTCENSAL	2,020	559	0.9993	3.09
ACS5YR	1,704	214	0.9996	34.44

grid cells that cover the same area. In some applications, the methods for determining these rules, or the rules themselves, are available, while in other products, such as Landsat [126, 127], the methodology is not public, and therefore, subsequent years of predictions are not comparable. Additionally, while some basic dasymetric heuristics, such as "humans do not live on land where the slope is over $45°$", can be globally applied, more detailed heuristics might be region specific. Our methodology seeks to bypass these potential problems as it only considers satellite imagery as input, therefore all of the predictions made by our model will be able to be explained in terms of the features of the input image. Similarly, because our models generate the probability that a section of satellite imagery belongs to each population class, we are able to show how confident our models are about a certain classification. We show these two components of our methodology in Figures 2.19 and 2.20 respectively.

In Figure 2.19 we show, for each class, the top 8 satellite image inputs from the testing set, that maximize the softmax output for that class. These images give us an insight into what types of features our model is learning. There are clear patterns moving from the lower classes, which represent sparsely populated areas, to very the upper classes which represent more urbanized areas. In the lower classes, most of the images contain some sort of roadway or distinctively marked fields. In classes 6 through 9 there are several buildings and developments visible, while finally in classes 10 through 14 there are dense suburban and urban developments with gridded patterns visible. In Figure 2.20 we show maps for several of the output population classes that show the estimated probability of each pixel

belonging to the respective class. From these we observe that our model makes confident predictions about the 0 population class (Layer 0), and the higher population classes. The lack of confidence in the lower population classes (Layers 2 and 4) makes sense as we do not expect the visual difference between 1km^2 areas in which 4 and 16 people live to be large. To compound this, census block geographies are larger in low population rural areas, meaning that our disaggregated "ground truth" training data will be noisier in lower population areas.

Prediction Errors

Here we show some of the errors of our model. Through inspecting the pixel class errors, i.e., the true population class value in 2010 (disaggregated from the Census population counts) minus the predicted population class values, we noticed that our model is systematically over-predicting some large areas. In Figure 2.21 we show three of these cases: Oak Ridge National Laboratory in Oak Ridge, TN, Anniston Army Depot in Anniston, AL, and Walt Disney World in Orlando, FL. These locations all share the property of having many man-made structures and signals of human activity, without the "ground truth" labeling of a population count from the Census data. Walt Disney World has many structures that look similar to those in high population residential areas, and therefore will always be mis-classified by a model that only relies on satellite imagery as input. In these cases, a traditional dasymetric modeling approach to disaggregating population will have an advantage over our model, as such an augmented approach could easily incorporate layers describing army bases, amusement parks, and other large spatial structures that will *not* have populations living within their borders. Finally, these observations are further evidence that our model is generalizing and learning useful semantic content about the input images with which to make its prediction.

2.5.4 Discussion

Our goal in this work is to train convolutional neural networks to create high-resolution gridded population maps using only satellite imagery, then validate our model's predictions both quantitatively and qualitatively. We predict population counts in the continental US at a $0.01^\circ \times 0.01^\circ$ ($\approx$ 1km^2) resolution for 2010, after training on data from 2000. To evaluate and validate our models, we first aggregate the population predictions at the county level, and compare them to ground truth county population counts from the 2010 census. Our models perform well on the task of projecting county population, with the best model having a median absolute error of 4,642, and although they are not better than traditional county population projection methods used by the US Census, they are able to make reasonable predictions. Secondly, we show what the models have learned by creating maps that show the estimated probability of each cell belonging to a given class, and by visualizing the satellite image inputs for each class that our model is most confidently classifying. We observe that the most confident images for each class follow an expected pattern, whereby images of rural areas with small roads and fields are classified as low population cells, and gridded urban areas with dense housing are classified as high population cells. Finally we qualitatively explain some of the errors that our model is making in terms of noisy input data; for example, our model predicts that an army base in Anniston, Alabama is a high population area, even though the "ground truth" census data says that the area is unpopulated.

For future work we plan on extending our current methodology in several different ways. In terms of the CNN training process, there are several changes and experiments that we would like to try: experimenting with different loss functions and loss function weighting schemes that could take the ordinal nature of our classification problem into account. Currently we optimize the categorical cross entropy, which will not discriminate between "small" and "large" errors, i.e., the loss will not penalize misclassifying a label with true class 11, as a 10, more than it would penalize misclassifying the 11 as a 1. We

also would like to try training a model on the entire US; as this task has the potential to use over 8 million samples, this will bring entirely different challenges to the deep learning process. In terms of applying and evaluating the models, we would like to use these models to predict population counts in countries where censuses are not taken as often, and are not taken at as fine of a resolution as in the US. Similarly, we want to experiment with the trade-offs between ground truth data resolution and model accuracy to determine the limits of the applicability of these models. Finally, we would like to apply transfer learning methods to this problem such as investigating whether pre-training models on land-use classification tasks result in better predictions or whether directly predicting nighttime light intensities helps.

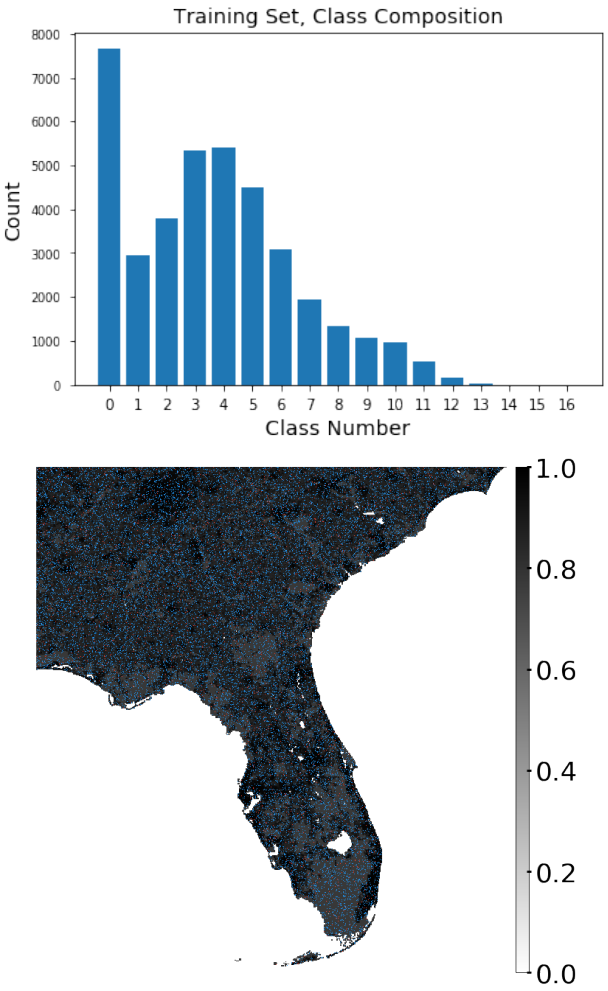


Figure 2.17: **Training/validation set sampling technique**. (Top figure) shows the counts of samples in the training set belonging to each of the target classes (i.e. the $C_t^{i,j}$ values). The target class values in the validation set follows the same distribution. (Bottom figure) shows the probability surface from which the training and validation points are sampled from; samples from the training set (38738 points) are shown in blue, and samples from the testing set (3874 points) are shown in red.

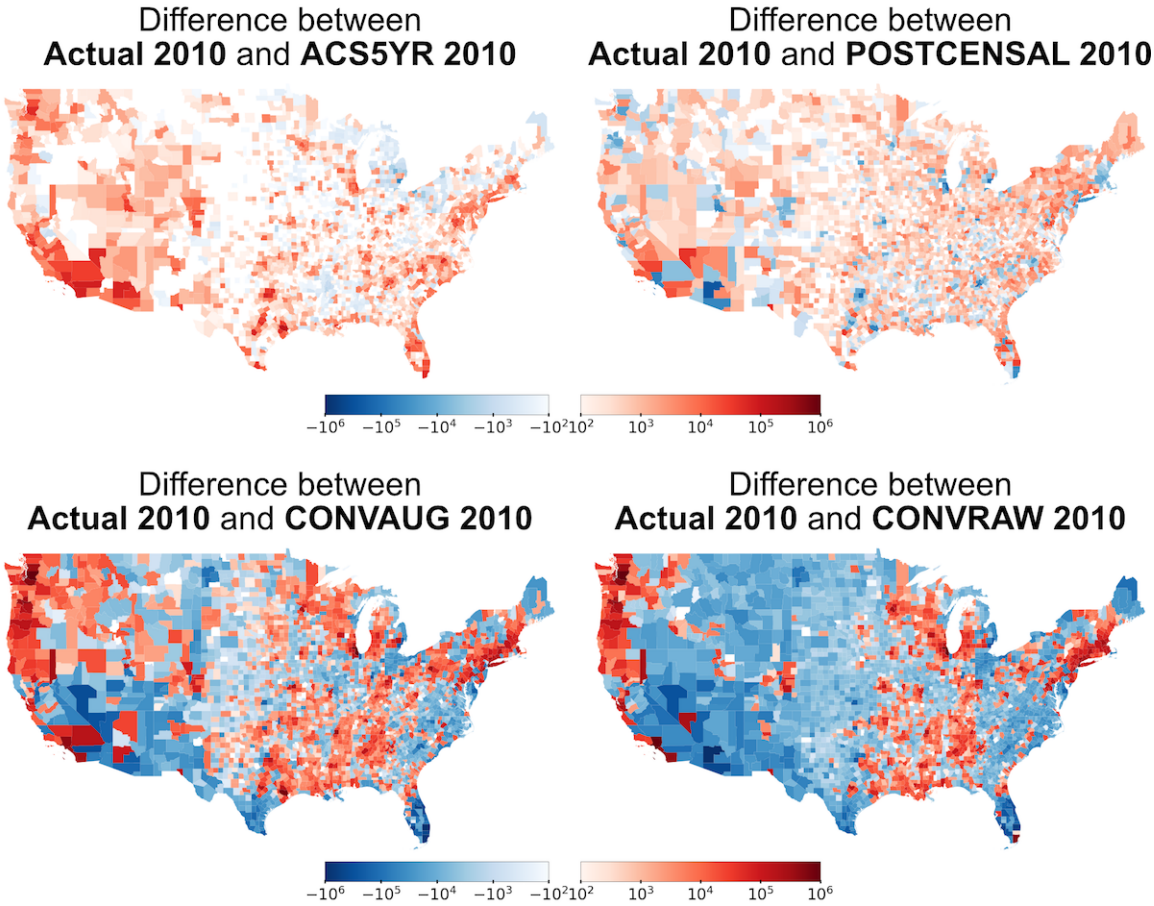


Figure 2.18: County level population projection results. Difference between the ground truth 2010 county population values and the tested methods for estimating county populations.

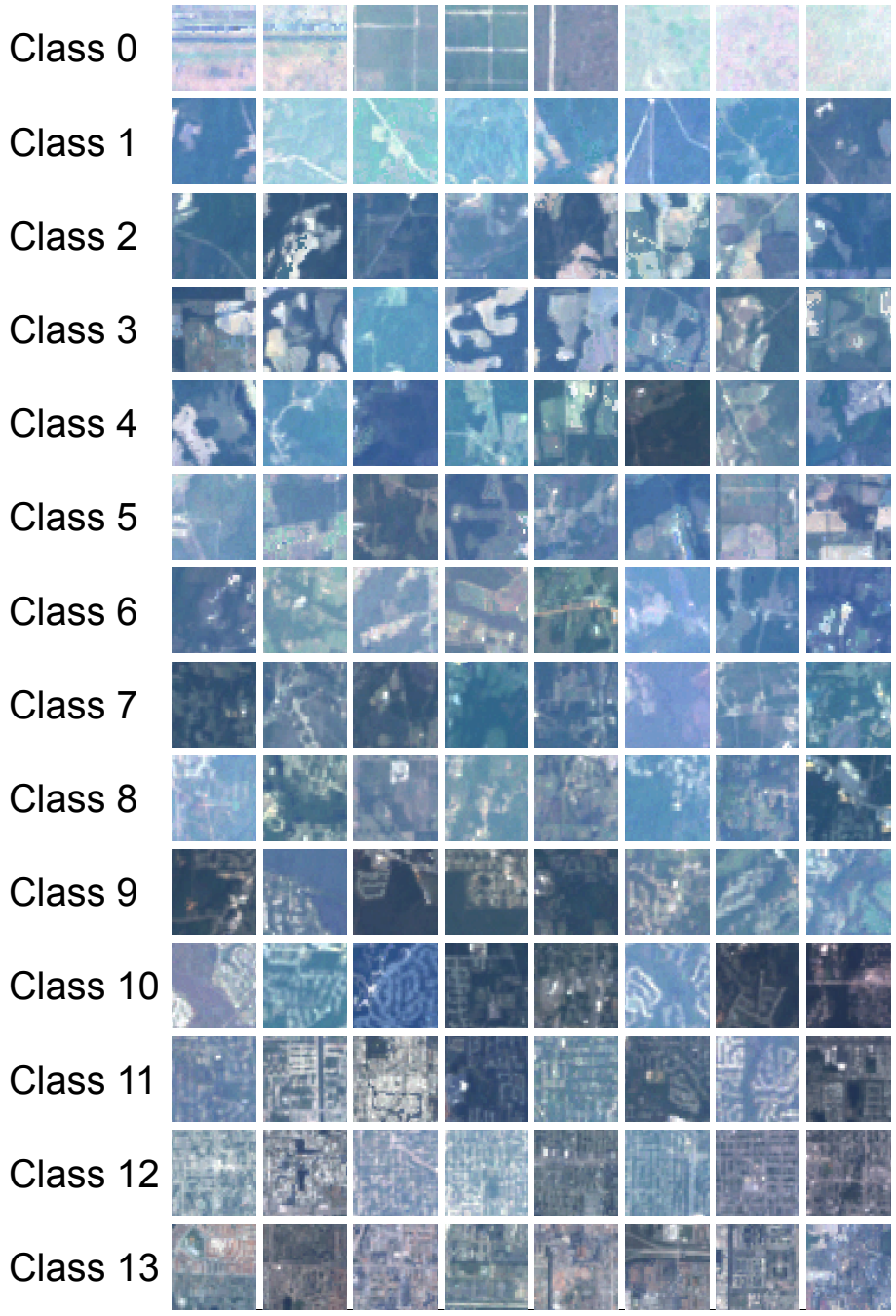


Figure 2.19: The top 8 most confident prediction images from the test set for each class (e.g. 99% prediction for a given class), all of which are correctly classified. Notice the types of images that appear from top (highways, few people) to bottom (buildings, many people) further indicated that our deep learning model is learning semantically-relevant features from satellite imagery.

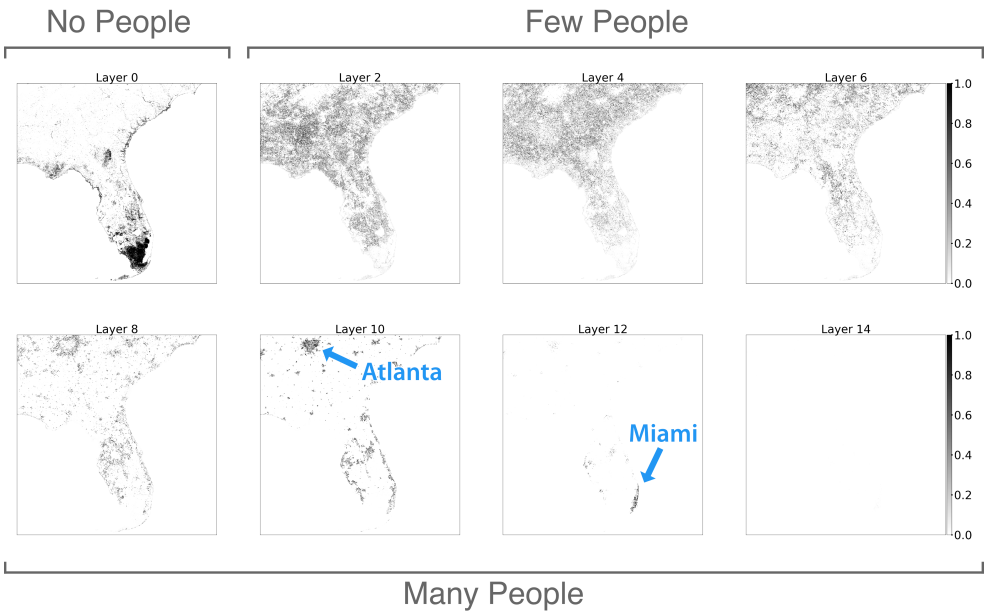


Figure 2.20: Activation maps for eight different population classes on the southeastern United States. Each map shows the estimated probability that a cell belongs in the map's population class. Layer 0 corresponds to zero people, layers 2, 4, and 6 correspond to few people, and layers 8, 10, 12, and 14 correspond to many people living in the activated areas. Notice the higher the layer number the more dense the population becomes, which naturally highlights urban cities such as Atlanta and Miami, annotated above.

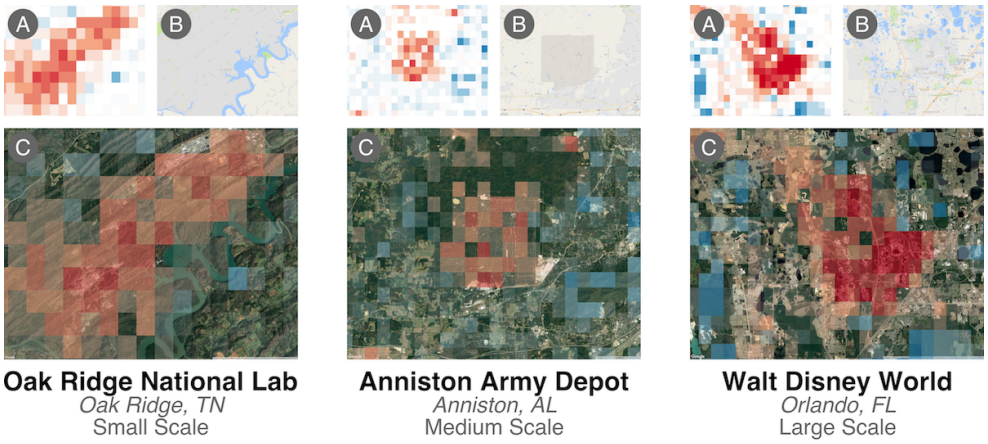


Figure 2.21: Three regions that have particularly high class prediction errors. Red pixels are over-predictions; blue pixels are under predictions. Upon inspection, these three regions are large-scale human-made areas that contain features typically associated with high-population areas, but in reality have very few people living in them. These include Oak Ridge National Lab (left, smaller scale), Anniston Army Depot (middle, medium scale), and Walt Disney World (right, large scale). (A) shows the class prediction errors, (B) shows the same region from Google Maps, and (C) shows (A) overlaid on the satellite imagery.

CHAPTER 3

MACHINE LEARNING IN HUMAN MIGRATION APPLICATIONS

3.1 Introduction

Large scale human migrations have had profound impacts throughout history. The "Great Migration" of African-Americans leaving the Southern United States during the beginning of the 20th century permanently changed the demographic and cultural landscape of the remainder of the country. The partition of British India into Pakistan and India in 1947 caused over 14 million people to migrate between the two newly formed countries, which set the stage for conflict between the two countries for the rest of the century. A more recent example is the "European migrant crisis", where, since 2014, millions of people from the Middle East and Africa have been legally and illegally migrating to European Union (EU) countries in an effort to escape war in their home countries. These migrants have been the source of political and social unrest, and ways of coping with the unprecedented numbers of migrants are only just being developed.

Human migration can be defined as the process through which people relocate themselves to new homes. Although the decision to migrate is largely personal and happens at the individual or family level for many reasons, large scale patterns in aggregate individual reasoning can be interpreted as driving forces behind migration. These driving forces are especially important to understand, as international migrants become more commonplace, and they have been traditionally broken down into "push" and "pull" factors [132]. In the first part of this Chapter we aim to learn these factors in a data-driven approach by modeling human migration with standard machine learning approaches. In the second part we couple migration models with sea level rise models to simulate human migration under different potential future scenarios.

3.2 Machine learning for human migration

Models of human mobility in their different forms are important for many reasons. Models of human commuting can help reduce traffic congestion and pollution, and can be used to drive land use policy and development choices [133]. Models of human migration are equally important to policy makers as they can give broader estimates of how the population of an area will change in upcoming years, how labor markets might be affected [134], how infectious diseases spread [135, 136], and how international trade will change [137]. Much recent research focuses on modeling human commuting flows [138, 11]; however little has focused on explicitly modeling human migration.

Human mobility has been traditionally modeled with the so-called gravity model, which posits that the probability of a trip between two locations decays directly as a function of the distance between them. This model was introduced in its modern form in 1946 [139] and has been used in many applications since [140, 141, 142, 143, 144, 145]. More recently, the radiation model [146] has been shown to capture long range trips better than gravity based models, and is described as 'a universal model for mobility and migration patterns'. The radiation model posits that the probability of a trip will decay indirectly with distance and directly with the amount of intervening opportunities, a notion first proposed by Stouffer [147]. The radiation model has been extended several times since being proposed in 2012 [138, 148, 149]. In general, gravity models have been shown to be more capable of reproducing commuting flows, i.e. human mobility at small spatial scales [11], while radiation models have been shown to be better at reproducing migration flows, i.e. human mobility at larger spatial scales [146]. Additionally, human migrations have been estimated by fitting generalized linear models derived from the gravity model [150, 151]. Both gravity and radiation models are analytical models with crafted functional forms and limited input data requirements. These models are focused on explaining human migration, rely on linear relationships between independent variables, and use hand crafted features

for each zone. These approaches, while useful for explaining human migration, trade predictive power for interpretability. Data sources such as the World Bank and US Census provide many zone-based features that can be algorithmically combined in a non-linear manner by tree or neural network based models to best predict human migration.

Our key contributions in this section are as follows:

1. We develop the first general machine learning formulation of the human migration prediction problem.

2. We develop a pipeline for training machine learning models to tackle this problem that includes procedures to deal with dataset imbalance, hyperparameter tuning, and performance evaluation.

3. We develop a custom loss function for training artificial neural networks that is more suitable for the migration prediction task.

4. We compare the performance of machine learning models to traditional models of human migration on two datasets, and show that the machine learning models outperform the traditional models in all cases.

3.2.1 Traditional migration models

In human mobility modeling, we are usually given n zones with the goal of predicting the number of people, T_{ij}, that move from every zone i to every other zone j. Traditional mobility models, such as the radiation model [146] and the gravity model [11], break the problem of estimating T_{ij} into two pieces: estimating the total number of people G_i that leave zone i (also referred to as a production function), and estimating the probability P_{ij} of a move occurring from i to j. The predicted number of migrants from i to j would be $\hat{T}_{ij} = G_i P_{ij}$. As a convention, the probabilities are normalized such that, given an origin i, the probabilities of traveling to all other destinations sum to 1, i.e. $\sum_{j=1}^{n} P_{ij} = 1$. If prior

information about the number of incoming and outgoing travelers per zone is known, then a constrained framework, such as the one described in Lenormand et al [11], can be used. If this information is not known, which is the case in predicting migrations, then a *production function* must be used to estimate the number of outgoing travelers per zone instead. A simple *production function* for a dataset can be found by expressing the number of outgoing migrants of a zone as a constant fraction of the population of that zone (for counties in the US, this percentage is ≈ 0.03 [146]). Given prior timestep's data on the population m_i and the corresponding number of outgoing migrants O_i in the current timestep for every zone i, the *production function* is expressed as $G_i = M(m_i) = \alpha m_i$, where α is the slope of the line of best fit through the pairs (m_i, O_i).

The traditional models of human mobility that we include in this study are: the radiation model [146], extended radiation model [148], and gravity models with both power and exponential distance decay functions [11]. The only information used by these models is: m_i, population of a zone i for both the origin and destination zones; d_{ij}, the distance between two zones i and j; and s_{ij}, a metric of intervening opportunities measured as the total population of all intervening zones between i and j, defined as all zones whose centroid falls in the circle centered at i with radius d_{ij} (not including zones i or j). See Table 3.1 for a description of each model.

Table 3.1: Traditional migration models

Model	**Equation**
Radiation	$\hat{T}_{ij} = M(m_i)\frac{m_i m_j}{(m_i+s_{ij})(m_i+m_j+s_{ij})}$
Extended Radiation	$\hat{T}_{ij} = M(m_i)\frac{[(m_i+m_j+s_{ij})^{\beta}-(m_i+s_{ij})^{\beta}](m_i^{\beta}+1)}{[(m_i+s_{ij})^{\beta}+1][(m_i+m_j+s_{ij})^{\beta}+1]}$
Gravity with Power Law	$\hat{T}_{ij} = M(m_i)\frac{m_i m_j d_{ij}^{-\beta}}{\sum_{k=1}^{n} m_k d_{ik}^{-\beta}}$
Gravity with Exponential Law	$\hat{T}_{ij} = M(m_i)\frac{m_i m_j e^{-\beta d_{ij}}}{\sum_{k=1}^{n} m_k e^{-\beta d_{ik}}}$

All of these models have two parameters that must be tuned using historical data: the production function parameter, α, that determines what fraction of the population of a zone

will migrate away in a given year, and a model parameter, β. From a socio-economic point of view, traditional models have the advantage of being interpretable, however we will show that this interpretability comes at a cost of predictive accuracy, as machine learning models can use similar historic data to achieve better results.

3.2.2 Evaluation methods

To evaluate how well alternative models perform, we use four metrics that compare how well a predicted migration matrix, $\hat{\mathbf{T}}$, recreates the ground truth values, $\mathbf{T}$. The first two of these (CPC, CPC_d) have been used in previous literature to evaluate human mobility models [10, 11], and the other two are standard regression metrics:

Common Part of Commuters (CPC) This metric directly compares numbers of travelers between the predicted and ground truth matrices. It will be 0 when the two matrices have no entries in common, and 1 when they are identical. We note that this metric, as used in previous studies of commuting flows, is identical to the Bray-Curtis similarity score used to compare abundance data in ecological studies [152, 153].

$$CPC(\mathbf{T}, \hat{\mathbf{T}}) = \frac{2\sum_{i,j=1}^{n} min(T_{ij}, \hat{T}_{ij})}{\sum_{i,j=1}^{n} T_{ij} + \sum_{i,j=1}^{n} \hat{T}_{ij}} \tag{3.1}$$

Common Part of Commuters Distance Variant (CPC_d) This metric measures how well a predicted migration matrix recreates trips at the same distances as the ground truth data. In this definition, N is a histogram where a bin N_k contains the number of migrants that travel between $2k-2$ and $2k$ kilometers. It will be 0 when the two matrices do not have any migrations at the same distance, and 1 when all fall within the same distances.

$$CPC_d(\mathbf{T}, \hat{\mathbf{T}}) = \frac{2\sum_{k=1}^{\infty} min(N_k, \hat{N}_k)}{\sum_{k=1}^{\infty} N_k + \sum_{k=1}^{\infty} \hat{N}_k} \tag{3.2}$$

Root mean squared error ($RMSE$) This is a standard regression measure that will "pun-

ish" larger errors more than small errors. This score ranges from 0 in a perfect match, to arbitrarily large values as the predictions become worse.

$$RMSE(\mathbf{T}, \hat{\mathbf{T}}) = \sqrt{\frac{1}{n}\sum_{i,j=1}^{n}(T_{ij} - \hat{T}_{ij})^2} \tag{3.3}$$

Coefficient of determination (r^2) This score measures the goodness of fit between a set of predictions and the ground truth values. This score ranges from 1, in a perfect fit, to arbitrarily negative values as a fit becomes worse, and is 0 when the predictions are equivalent to the expectation of the ground truth values.

$$r^2(\mathbf{T}, \hat{\mathbf{T}}) = 1 - \frac{\sum_{i,j=1}^{n}(T_{ij} - \hat{T}_{ij})^2}{\sum_{i,j=1}^{n}(T_{ij} - \bar{T})^2} \tag{3.4}$$

In addition to the previous four metrics, we compare the ground truth number of incoming migrants and the predicted number of incoming migrants per zone using mean absolute error (MAE) and r^2. The predicted number of incoming migrants for a zone, i, is calculated as $\hat{v}_i = \sum_{j=1}^{n} T_{ji}$. We argue that it is important to explicitly measure how well each model performs at estimating the number of incoming migrants, because the number of incoming migrants to a location will be the most important measure for policy makers in that area. Incoming migrant predictions can inform population growth estimates and hence infrastructure planning and job analysis.

3.2.3 Learning migration models

Formally, the problem of modeling human migration is as follows: given n zones, d_1 features describing each zone, $\mathbf{F} \in \mathbb{R}^{n \times d_1}$, and d_2 joint features describing features between a pair of zones, $\mathbf{J} \in \mathbb{R}^{n \times n \times d_2}$, at some timestep t, the objective is to predict the origin/destination *migration* matrix $\hat{\mathbf{T}} \in \mathbb{N}^{n \times n}$ at the next timestep, $t + 1$, where an entry $\hat{T}_{ij}$ represents the estimated number of migrants relocating from zone i to zone j. Our goal

is to estimate a function $f(F_{i:}, F_{j:}, J_{ij}) = \hat{T}_{ij}$, which takes the features of zone i and j, as well as the joint features between them, as input, and directly outputs the estimated number of migrants that travel from i to j. This approach is different from how the traditional migration models work as it does not require a *production function*, but instead directly predicts $\hat{T}_{ij}$. This formulation contains the simplifying assumption that migrant flows are static in time, meaning that they can be entirely determined by the features from the previous timestep. In reality migrant flows will be dependent on temporal features, such as long term developmental trends, however many places will not have enough data to take advantage of these patterns. With this formulation, our models can be applied more broadly to predict future migration patterns in locations that have only collected a single year of data.

Hyperparameter optimization

To fit f for a given dataset, we will train two machine learning models, "extreme" gradient boosting regression (XGBoost model) [154], and an artificial neural network model (ANN model) [155]. Each of these models contains several hyperparameters that must be tuned to obtain good performance on a given learning task. Our first model, the XGBoost model, is a standard machine learning model based on gradient boosting trees [156] that often performs very well on many regression and classification tasks. One benefit of this model is that it gives a ranking of the relative feature importances [157]. The parameters of the XGBoost model that we consider for hyperparameter tuning are the maximum tree depth, number of estimators, and learning rate. Our ANN model is composed of densely connected layers with ReLU activation layers[1]. We tune the following ANN parameters: loss function, number of layers, layer width, number of training epochs, and training mini-batch size.

[1]Our model is implemented in Python with the Keras library: `https://keras.io/`

Dealing with zero-inflated data

We observe that migration data is heavily zero-inflated, where in any given year, most pairs of zones do not have any migrants traveling between them, i.e. $T_{ij} = 0$ for most (i, j) pairs. Considering migrations between US counties [158], less than 1% of the possible pairs of counties have migrations between them. This imbalance will cause problems for machine learning models. To address this problem, when creating a training dataset we undersample "negative" samples between pairs of zones for which there are no observed migrations. This is a naïve technique that will necessarily throw out available information [159]. To offset this, we introduce a hyperparameter k that determines the number of "negative" examples of migrations to train with. If there are n_t pairs of zones where there are observed migrations, "positive examples", we include all n_t, and an additional $n_t k$ randomly chosen zone pairs where there are no observed migrants. This hyperparameter is included in both the XGBoost and ANN model parameter searches. We give further details on the hyperparameter tuning process in Section 3.2.4.

Custom ANN loss function

Previously we mentioned that our ANN model will consider different loss functions as a hyperparameter. Common loss functions for regression tasks include "mean squared error" (MSE), "mean absolute error" (MAE), and "mean absolute percentage error" (MAPE). Our preliminary experimental results show that MAE and MAPE loss functions perform poorly, partially due to their inability to punish large errors and deal with many zeros respectively. To contrast with the aforementioned zero-inflation problem, we observe that the distribution of migrant counts has a long tail, whereby few pairs of zones consistently experience large volumes of migrations. Considering these observations, and because the CPC metric is one of the key metrics of interest (described in detail in Section 3.2.2), we derive a new loss

function based on CPC to train the ANN model with. This loss function is given as:

$$L(y, \hat{y}) = 1 - \frac{2\sum_{i=1}^{n} min(y_i, \hat{y}_i)}{\sum_{i=1}^{n} y_i + \sum_{i=1}^{n} \hat{y}_i} \tag{3.5}$$

Where y_i is a migration flow entry from $\mathbf{T}$. The gradient update for this loss function is:

$$\frac{\partial L(y, \hat{y})}{\partial \hat{y}_j} = \frac{2\sum_{i=1}^{n} min(y_i, \hat{y}_i)}{(\sum_{i=1}^{n} y_i + \sum_{i=1}^{n} \hat{y}_i)^2} - \frac{\begin{cases} 2 & \hat{y}_j < y_j \\ 0 & \text{otherwise} \end{cases}}{\sum_{i=1}^{n} y_i + \sum_{i=1}^{n} \hat{y}_i} \tag{3.6}$$

Intuitively, this loss will 'reward' predictions where the number of migrants matches the ground truth, while also enforcing per link absolute error to be minimized. This loss function is not exactly equivalent to the CPC, as during the ANN training it will only consider a single mini-batch worth of samples at a time (in our case $|y| = |\hat{y}| = |\text{mini batch}|$, where as the CPC metric is a function of the entire migration matrix). Empirically this custom loss function results in better validation performance and faster training times than MSE loss.

3.2.4 Experiments

Datasets

We perform experiments comparing the performance of traditional models to our machine learning models on two datasets, the *USA Migration* dataset and the *Global Migration* dataset.

The *USA Migration* dataset consists of yearly intra-county migrations in the USA between 3106 counties from the IRS Tax-Stats data [158] for the 11 years in the range from 2004 to 2014. We supplement the migration data with the following 7 per-county features (taken from the US Census estimates and calculated from the Census TIGER line maps of US county boundaries): population, land area, population density, median household

income, county water area, is a coastal county, and number of neighboring counties. In addition to these 7 per-county features, we add the following between-county features: distance, intervening population, intervening land area, intervening number of counties, intervening population density, and intervening median income. The intervening features are calculated based on the idea of "intervening opportunities" presented in the radiation model. For any given county-level variable, x, e.g. population, the intervening amount of that variable between counties i and j is defined as $s^x_{i,j}$, the sum of all x in the intervening counties that fall within the circle centered at county i with a radius to county j (excluding x_i and x_j).

The *Global Migration* dataset consists of decadal inter-country migration data between 207 countries from the World Bank Global Bilateral Migration Database [160]. The *Global Migration* dataset contains 5 timesteps, one every 10 years from 1960 to 2000. In the *Global Migration* dataset we add the following 5 per-country features (taken from World Bank World Development Indicators data [161]): population, population density, population growth, agricultural emissions, and land area. Additionally, we include 3 between-country features: distance, intervening population, and intervening land area.

For each year of data in the *USA Migration* and *Global Migration* datasets we create an 'observation' for each pair of zones, an origin zone and destination zone (counties in the *USA Migration* dataset and countries in the *Global Migration* dataset). Each observation consists of the per-zone features for both the origin zone and destination zone (population of origin, population of destination, etc.) and the between-zone features of the origin and destination. This corresponds exactly to the $F_{i,:}, F_{j,:}, J_{ij}$ of the function f (described in Section 3.2.3) that we want to learn. An observation is associated with the target number of migrants, T_{ij}, traveling from the origin to the destination. Formally, for a given year, t, number of zones, n, number of per-zone features, d_1, and number of between-zone features, d_2, we create a matrix of observations $\mathbf{X}_t \in \mathbb{R}^{n^2 \times (2d_1+d_2)}$ and vector of targets $Y_t \in \mathbb{R}^{n^2}$, capturing the migration flows observed in year $t+1$.

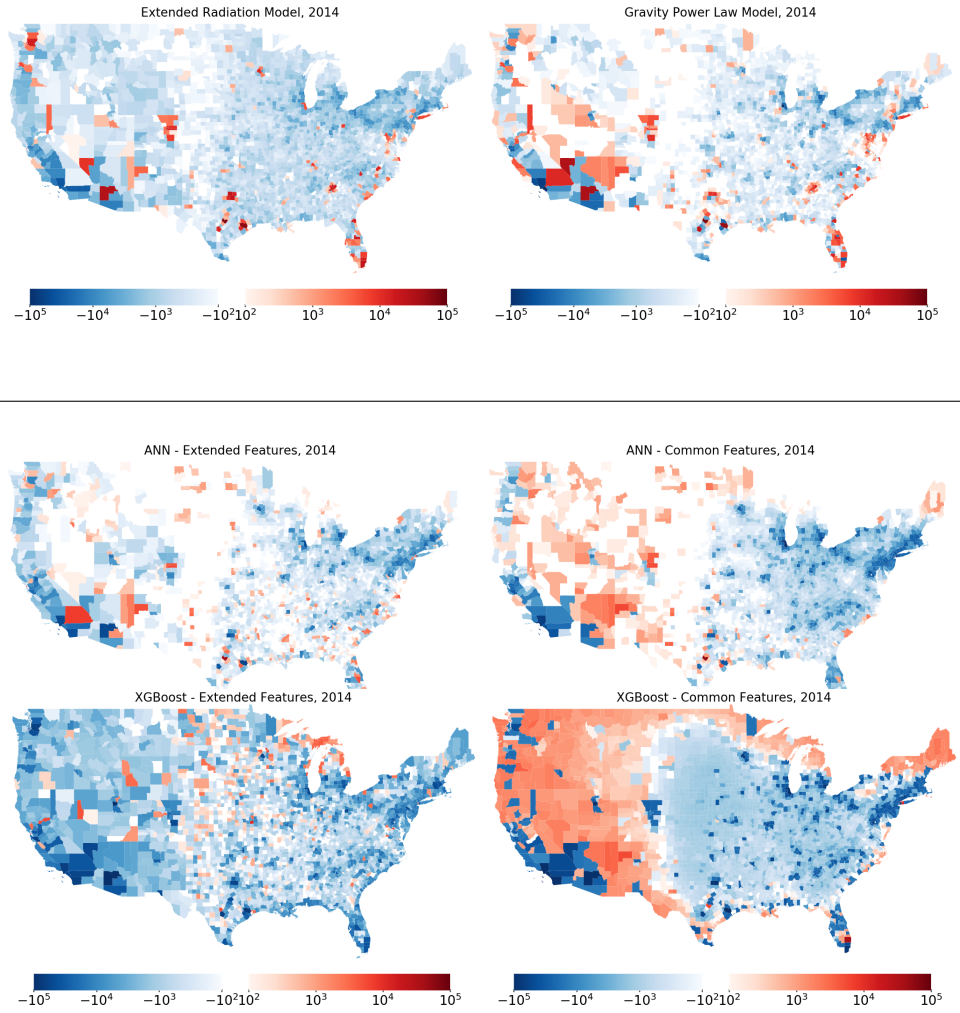


Figure 3.1: ***USA Migrations* modeling error.** These maps show the difference between the ground truth number of incoming migrants and predicted number of incoming migrants per county for 6 models in 2014. Blue corresponds to overestimation by the model, red to underestimation by the model, and white if the model accurately predicts the correct number of incoming migrants. **Top row** shows the results for the Extended Radiation model and Gravity model with power law distance decay. **Middle row** shows the results for ANN models trained with the extended and common feature sets. **Bottom row** shows the results for XGBoost models trained with the extended and common feature sets.

Experimental Setup

To select the hyperparameters of the models that we described in Sections 3.2.1 and 3.2.3, we consider triplets of "years" of data as training, validation, and testing sets. Specifically, for three years of data $\{(X_{t-2}, Y_{t-2}), (X_{t-1}, Y_{t-1}), (X_t, Y_t)\}$, we call (X_{t-2}, Y_{t-2}) the training set, (X_{t-1}, Y_{t-1}) the validation set, and (X_t, Y_t) the test set. We tune the hyper-

Table 3.2: *USA Migration* results. Comparison of the ANN and XGBoost models with and without a production function to traditional migration models. The values shown in the table are the average and standard deviations of the models' test performance on 2006 through 2014 data. Bold values indicate the best values per column.

USA Migrations	**Metrics on full matrix**				**Metrics on incoming migrants vector** (Average Incoming Migrants = 3,196)	
Production Function	CPC	CPC_d	$RMSE$	r^2	MAE	r^2
Gravity Model Exponential Decay	0.53 +/- 0.01	0.66 +/- 0.02	87.4 +/- 9.0	-1.48 +/- 0.28	1,216 +/- 128	0.67 +/- 0.03
Gravity Model Power Law Decay	0.56 +/- 0.01	0.78 +/- 0.02	75.7 +/- 8.0	-0.86 +/- 0.26	1,129 +/- 129	0.72 +/- 0.04
Radiation Model	0.53 +/- 0.01	0.76 +/- 0.02	47.6 +/- 5.0	0.26 +/- 0.09	1,346 +/- 148	0.80 +/- 0.02
Extended Radiation Model	0.58 +/- 0.01	0.83 +/- 0.01	35.6 +/- 3.0	0.59 +/- 0.03	1,123 +/- 117	0.86 +/- 0.02
XGBoost model - traditional features	0.51 +/- 0.08	0.74 +/- 0.07	28.6 +/- 5.4	0.72 +/- 0.10	1,151 +/- 249	0.86 +/- 0.04
ANN model - traditional features	0.63 +/- 0.01	0.86 +/- 0.02	35.1 +/- 3.2	0.60 +/- 0.03	911 +/- 107	0.91 +/- 0.01
XGBoost model - extended features	0.58 +/- 0.03	0.78 +/- 0.02	24.2 +/- 1.4	0.81 +/- 0.02	968 +/- 56	0.89 +/- 0.02
ANN model - extended features	0.68 +/- 0.01	0.89 +/- 0.02	29.8 +/- 2.7	0.71 +/- 0.02	935 +/- 98	0.91 +/- 0.02
No Production Function	CPC	CPC_d	$RMSE$	r^2	MAE	r^2
XGBoost model - traditional features	0.54 +/- 0.11	**0.99 +/- 0.02**	18.5 +/- 6.1	0.88 +/- 0.08	3,091 +/- 1,740	0.41 +/- 0.85
ANN model - traditional features	0.63 +/- 0.02	0.88 +/- 0.06	35.3 +/- 3.5	0.60 +/- 0.04	1,188 +/- 259	0.84 +/- 0.16
XGBoost model - extended features	0.62 +/- 0.04	**0.99 +/- 0.02**	**13.0 +/- 1.5**	**0.94 +/- 0.02**	2,060 +/- 622	0.76 +/- 0.28
ANN model - extended features	**0.69 +/- 0.01**	0.93 +/- 0.05	28.0 +/- 3.6	0.75 +/- 0.03	**909 +/- 48**	**0.92 +/- 0.04**

parameters of the models using a randomized grid search with 50 sampled hyperparameters using the training and validation sets. We select the best set of hyperparameters according to the CPC score, then use those parameters to train a model on the validation set and record its performance on the test set. We repeat this process for each $(t-2, t-1, t)$ triplet of years in each dataset. Our final results are reported as averages over the test set results.

A hyperparameter present in both XGBoost and ANN models is the downsampling rate, k. As a preprocessing step for a given year of training data $(\mathbf{X}_t, Y_t)$, we include all observations, X_i where $Y_i > 0$ (let this number of samples be m), and choose $k*m$ random samples with replacement from the remaining observation (where $Y_i = 0$). This sampling process only takes place when training a model. When testing a model on the validation or test sets, the full datasets are always used. For experiments with the *USA Migration* dataset we consider values of k in a uniform distribution of integers from 5 to 100, while for experiments with the *Global Migration* dataset we consider the uniform distribution of integers from 1 to 5, because the average percentage of non-zeros is $< 1\%$ and 20% respectively.

For the XGBoost models, we sample the following parameters: maximum tree depth

Table 3.3: *Global Migration* results. Comparison of the ANN and XGBoost models with and without a production function to traditional migration models. The values shown in the table are the average and standard deviations of the models' test performance on 2006 through 2014 data. Bold values indicate the best values per column.

Global Migrations	**Metrics on full matrix**				**Metrics on incoming migrants vector** (Average Incoming Migrants = 674,858)	
Production Function	CPC	CPC_d	$RMSE$	r^2	MAE	r^2
Gravity Model Exponential Decay	0.16 +/- 0.00	0.16 +/- 0.00	62,218 +/- 5,341	0.02 +/- 0.03	651,194 +/- 80,220	0.00 +/- 0.02
Gravity Model Power Law Decay	0.16 +/- 0.00	0.15 +/- 0.00	61,523 +/- 5,278	0.05 +/- 0.00	628,678 +/- 79,474	0.03 +/- 0.03
Radiation Model	0.16 +/- 0.00	0.14 +/- 0.00	62,173 +/- 5,277	0.02 +/- 0.00	614,483 +/- 79,378	0.04 +/- 0.02
Extended Radiation Model	0.16 +/- 0.00	0.14 +/- 0.00	62,108 +/- 5,299	0.03 +/- 0.00	618,576 +/- 76,150	0.03 +/- 0.02
XGBoost model - traditional features	0.18 +/- 0.01	0.14 +/- 0.01	58,377 +/- 5,141	0.14 +/- 0.01	597,478 +/- 79,178	0.10 +/- 0.01
ANN model - traditional features	0.19 +/- 0.01	0.15 +/- 0.01	60,272 +/- 4,610	0.08 +/- 0.02	589,789 +/- 79,542	0.26 +/- 0.03
XGBoost model - extended features	0.21 +/- 0.01	0.16 +/- 0.01	57,909 +/- 5,409	0.16 +/- 0.02	573,090 +/- 56,987	0.15 +/- 0.02
ANN model - extended features	0.22 +/- 0.02	0.17 +/- 0.01	58,887 +/- 4,477	0.12 +/- 0.02	563,259 +/- 74,127	0.24 +/- 0.06
No Production Function	CPC	CPC_d	$RMSE$	r^2	MAE	r^2
XGBoost model - traditional features	0.33 +/- 0.02	0.59 +/- 0.03	52,729 +/- 5,455	0.26 +/- 0.26	938,905 +/- 172,834	0.18 +/- 0.28
ANN model - traditional features	0.33 +/- 0.01	0.37 +/- 0.04	56,005 +/- 882	0.20 +/- 0.11	537,351 +/- 44,034	**0.53 +/- 0.16**
XGBoost model - extended features	**0.43 +/- 0.03**	**0.64 +/- 0.02**	**47,329 +/- 5,073**	**0.42 +/- 0.13**	577,473 +/- 77,315	0.48 +/- 0.34
ANN model - extended features	0.40 +/- 0.02	0.43 +/- 0.02	50,921 +/- 3,556	0.33 +/- 0.13	**459,841 +/- 55,479**	0.52 +/- 0.30

from $\mathcal{U}\{2, 7\}$[2], number of estimators from $\mathcal{U}\{25, 275\}$, and learning rate uniformly in the range from 0 to 0.5. For the ANN models we sample the following parameters: network loss function uniformly from {'CPC Loss', 'MSE'}, number of layers uniformly from $\mathcal{U}\{1, 5\}$, layer width from $\mathcal{U}\{16, 128\}$, number of training epochs from $\mathcal{U}\{10, 50\}$, and training mini-batch size uniformly from $\{2^9, 2^{10}, 2^{11}, 2^{12}, 2^{13}, 2^{14}\}$.

We calibrate the parameters of the traditional models in a similar manner. Every traditional model, except for the radiation model, has two parameters, α and β, that must be calibrated to give useful results (where the radiation model only uses α). For each pair of years of data, $\{(X_{t-1}, Y_{t-1}), (X_t, Y_t)\}$, that we refer to as training and testing sets respectively, we find the value of α that gives the best production function on the training set. Similarly, we find the value of β that maximizes the CPC score of each traditional models on the training set. We then use these α and β values to run each model on the testing set, and report the results as averages over all test set results.

To directly compare how the ML models and traditional models perform under the same conditions, we perform experiments where the ML models are used with the same production functions as the traditional models. This imposes an artificial constraint on

[2] $\mathcal{U}\{a, b\}$ is the discrete uniform distribution.

Table 3.4: Top 10 most important (extended) features in both the *USA Migration* and *Global Migration* datasets. The values in the table show the average and standard deviations of the information gain feature importances from an XGBoost model trained on the extended feature set for each timestep of data.

USA Features	**Importance**	**Global Features**	**Importance**
Intervening number of counties	25.3% +/- 2.4%	Population growth of origin	19.5% +/- 16.0%
Population of origin (trad)	15.7% +/- 1.7%	Intervening population (trad)	12.3% +/- 3.7%
Population of destination (trad)	14.2% +/- 0.9%	Agricultural emissions of destination	10.6% +/- 5.8%
Intervening population (trad)	6.1% +/- 1.2%	Intervening land area	8.7% +/- 5.6%
Is destination coastal	4.3% +/- 4.6%	Population growth of destination	7.9% +/- 6.2%
Distance between counties (trad)	3.7% +/- 0.9%	Population of destination (trad)	6.9% +/- 0.8%
Intervening area	3.6% +/- 0.9%	Distance between counties (trad)	6.6% +/- 1.9%
Area of destination	3.5% +/- 0.5%	Population of origin (trad)	6.1% +/- 1.6%
Number of neighbors destination	3% +/- 1.6%	Population density of destination	5.7% +/- 4.5%
Water area of origin	3% +/- 1.3%	Land area of origin	5.2% +/- 3.1%

the ML models, as these models are able to directly estimate the number of migrations between two zones, without supplemental information on the number of outgoing migrants from each zone. To apply a production function, M, to the predictions made by a ML model, $\hat{\mathbf{T}}$, we create a new set of predictions, $\hat{\mathbf{T}}'$, where an entry $\hat{T}'_{ij} = M(m_i)\frac{\hat{T}_{ij}}{\sum_{k=1}^{n}\hat{T}_{ik}}$.

Results

Tables 3.2 and 3.3 show the average results over all years of data of the ML models and traditional models in the *USA Migration* and *Global Migration* datasets respectively. From these tables we observe that the best traditional model for the *USA Migration* dataset is the Extended Radiation model, beating the other traditional models in all metrics. None of the traditional models are able to capture the migration dynamics in the *Global Migration* dataset; they all have an r^2 score near 0, meaning that a model which predicts the average number of migrants for every link would perform just as well. The ML models perform much better. In the case where the ML models are constrained to the same conditions as the traditional models, using traditional features and a production function, the ANN model beats all of the traditional models in 5 out of the 6 measures in the *USA Migration* datasets, and the XGBoost model beats all the traditional models in 4 out of the 6 metrics.

Similarly, the ML models considerably outperform the traditional models in the *Global Migration*, outperforming them in all metrics. Considering the extended feature set results, the ML models perform even better. The ANN and XGBoost models without a production function outperform the same models with a production function in 5 out of the 6 metrics. The XGBoost model outperforms the ANN model in 3 out of the 4 metrics that evaluate the models' per link predictions, however the ANN model performs better on the two metrics that evaluate the aggregate incoming migrant prediction performance.

These results suggest that more features than those which are used by the traditional models, are needed to accurately predict human migrations. The ability of ML models to incorporate any number of additional features is one of the key motivations for using them to obtain more accurate results. Considering this, it will be insightful to understand, which of the features are most informative to the ML models. Since feature importance analysis for ANNs is quite challenging, we report in Table 3.4 the top 10 most important features for both datasets (based on information gain) in the XGBoost model trained on the *extended feature set*, averaged over all years of data. In both datasets, the intervening population feature is in the top 4 important features which validates the intuition that intervening opportunities are important in predicting migrations. The most important feature in the *USA Migration* dataset is the number of intervening counties between two locations, a simpler form of the intervening population idea. In the *Global Migration* dataset, the population growth of the origin is the most important feature on average, with a large standard deviation. In some years this feature is very important, however in other years it is less so. Intuitively, population growth will be correlated with the amount of incoming migration. During relatively stable years, with small population growth, other features will be more predictive of migration.

In Figure 3.1 we show the difference between the actual and predicted numbers of incoming migrants per county for the two best traditional models, and all of the ML models without production functions. From these maps we can see that between ML models, those

trained with the extended feature set perform better than those trained with only the traditional features. Specifically, without the extended features, the ML models underpredict the number of migrations to the western portion of the United States. When the extended features are taken into account, the models are able to correct for this spatial bias. Holding with the experimental results, we can see that the ANN model with extended features best captures the incoming migrant distributions per county. The ANN model is able to more accurately match the number of migrants that travel to rural areas (e.g. to the midwestern US), compared to the traditional models that consistently over estimate the numbers of migrants to rural areas. In general, these maps agree with our empirical results, that the ANN model (with the lowest average incoming migrants MAE) is able to best predict migrations.

3.2.5 Discussion

With the increasing availability of high resolution socio-economic data in countries that record human migrant flows, it is possible to use machine learning models of human migration rather than traditional gravity or radiation models. Machine learning models offer greater levels of modeling flexibility, as they can combine many input features in non-linear ways that can not be captured by static equations. Furthermore, machine learning models can be easily customized to the problem or country at hand.

We develop two machine learning based models for the task of predicting human migration flows, for both between counties in the US and between countries across the world. We compare these models to traditional human migration models using two sets of features and show that our models outperform the traditional models in most of the evaluation metrics.

We would like to extend this work to better explain human migration through a more complete analysis of features included in the model, and incorporating different models. While the XGBoost model can provide a ranking of feature importances, this does not

fully explain the dynamics that drive human migration. Additionally we would like to study how these migration models could be specialized to predict migrations under extreme weather events. Hurricanes and other natural disasters can displace large populations, and determining where these populations will resettle is the first step towards providing an unique planning tool for policy makers. To achieve these goals, higher resolution migration data, on both spatial and temporal scales, will need to be obtained. Extreme weather events, by definition, are short lived and their effects are averaged out in coarser resolution datasets such as the ones used in this study. Finally, we would like to study the connections between human migration and other processes primarily driven by aggregate human behaviors such as inter- and intra-national trade.

3.3 Migration patterns under different scenarios of sea level rise

Climate change will affect millions of people around the world. Human migration is a natural response to these climate change pressures, and is one of many adaptation measures that people will take in response to climate change [162, 163, 164, 165]. Understating how human migration will be affected by climate change is therefore a critical input in the decision making process of many governments and organizations. In particular, it is important to understand how climate change driven migration will differ from "business as usual" forms and motivations humans have to migrate. Yet, an empirical assessment of the process remains elusive. In this paper, we propose a framework that recognizes the imperfections of any given assessment regarding migration, but allows us to think critically about the possible ways climate change can alter migration patterns.

In particular, our framework assesses the broader impacts of climate change on population, by explicitly considering the effects on migration on populations directly affected by climate change and *indirectly affected* by the change in migration patterns induced by climate change. The framework we propose here is not intended to explain individual decisions, but to characterize aggregate patterns that can nonetheless help prepare policy

responses by local communities and governments.

Accounting for indirectly affected people is one of the main contributions of our framework; these are people that live in locations that experience increased population pressures due to heightened inflows of climate migrants. These indirect effects will cause accelerated changes for inland areas, particularly urban areas, that will observe much higher levels of incoming migrants than they would have without climate impacts. These changes can in turn take the form of tighter labor markets [166] and increased housing prices [167], with broader effects on income inequality in the coastal areas [168]. Of course, migration to other cities can also have positive impacts; new migrants can improve productivity as they bring with them human capital accumulated elsewhere [165]. Thus, explicitly accounting for *indirect effects*, even under large uncertainties, is an important part of quantifying the effects of climate change.

Broadly speaking, migration processes can be characterized by three components: sources, destinations, and flows between them. Climate change will affect each of these components in different ways. For example, increased climate burden on agricultural regions can increase migrants to move to more urban spaces or to move to different towns, provinces or even different countries. Climate change can also induce conflict, thus increasing the number of refugees. Climate change can also affect destinations, for example by making cities less livable due to urban heat island effects [169] or due to increased burden on services such as water and electricity [170]. By affecting both the origin and destination, climate change also affects the flow of migrants. As more research is conducted about different aspects of the migration process, our framework—by *making explicit the different patterns of migration* (climate change driven or "business as usual")— will allow to incorporate scientific knowledge about changes across all these aspects of the migration process.

We introduce and discuss our framework in the context of sea level rise impacts on human migration. Sea level rise (SLR) will affect millions of people living in coastal areas. Different studies have highlighted likely scenarios of sea level rise by 2100, varying in

their projections of severity. According to the IPCC Fifth Assessment Report, in the "worst-case" Representative Concentration Pathways (RCP) scenario, RCP 8.5, where greenhouse gas emissions continue to rise throughout the 21st century, a global mean sea level (GMSL) rise between 0.52 to 0.98 meters (m) is likely by 2100 [171]. Other estimates, using statistical instead of process based models of GMSL, project a rise in the range of 0.75 m to 1.9 m by 2100 [172]. Recent research from the National Oceanic and Atmospheric Administration (NOAA) has even suggested a 2.5 m upper bound of GMSL rise by 2100 for an 'extreme' SLR scenario, and a 2 m GMSL rise for a 'high' scenario [173].

The impacts of SLR are potentially catastrophic. About 30% of the urban land on earth was located in high-frequency flood zones in 2000, and it is projected to increase to 40% by 2030 taking urban growth and SLR into account [174]. In the United States alone, 123.3 million people, or 39% of the total population, lived in coastal counties in 2010, with a predicted 8% increase by the year 2020 [175]. By the year 2100, a projected 13.1 million people in the United States alone would be living on land that will be considered flooded with a SLR of 6 feet (1.8 m) [176].

As oceans expand and encroach into previously habitable land, affected people - climate migrants - will move towards locations further inland, looking for food and shelter in areas that are less susceptible to increased flooding or extreme weather events. In this paper, we argue that the comprehensive impacts of SLR on human populations, when considering migration, expand far beyond the coastal areas.

While discussions regarding SLR impacts on human populations are often constrained to regions *directly* experiencing SLR-driven flooding [177, 178, 176], several studies have investigated the connection to climate driven human migration. Several theoretical frameworks use qualitative case studies to motivate models that might represent the reasoning behind migration choices due to SLR, but are not grounded in statistical methods [162, 179, 180]. There are many complex interactions between demographic driven migration and climate change driven migration, and the scope and scale of the impacts of climate

change on migration will be significant [163, 181]. One example of these impacts that has been studied considers the political ramifications that will come with the eventual migrants from Pacific island of Kiribati, which will most likely become completely flooded under a 3 meter SLR [182]. Another example is the projected widening demographic differentials in countries that will be especially impacted by SLR [181, 183], similar to the demographic changes seen after the 1970s droughts in Africa [184]. The foundation of both of these concerns is in people's destination locations, therefore it is prudent to weigh the question of 'where' people will go equally with 'how many' people will be initially affected [185].

There are also few empirical studies that link climate change with human migration patterns. Feng et al. show that the negative impacts of climate change on crop-yields has driven increased emigration from Mexico to the United States [163], while Thiede and Gray examine the effects of changing climate variables on the timing of migration in Indonesia [186]. The only empirical works that examine the *effects of SLR on human migration* do so by coupling population projections with SLR models and migration models to estimate how population distributions might change in future scenarios [176, 187, 188]. In the US, small area population projections for the year 2100 have been combined with spatially explicit estimates of SLR [176] and an unobserved component regression model to estimate the destinations of populations that could be forced to migrate through coastal flooding. In [187], approximately 56% of counties in the US are found to be affected by larger migrant influxes under 1.8 m of SLR. Similarly, in Bangladesh, gridded population projections have been combined with a "bathtub" type model of SLR and the radiation model of human migration to estimate how population distributions may change [188]. This coupled model has minimal data requirements, forecasts large quantities of immigration to the division of Dhaka in Bangladesh, and highlights the broader potential impacts of these migrants including an increased demand for housing, food, and jobs.

These empirical studies make the critical simplifying assumption that climate driven migration will follow the same patterns as historic migration . In fact, even though Hauer [187]

highlights that "climate migrants resulting from press stressors will probably constitute 'enhanced', or extra, normal out-migration", in the paper he assumes that migrations will happen only between locations for which there are historically observed migrations. However, human migration is a function of push and pull factors, and increased climate stress will affect both [179]. Our framework, by incorporating separate models of migration choices for climate change driven versus "business-as-usual" migration, recognizes that the patterns of climate migrants will not necessarily follow patterns observed in historical migration data. In our application, this distinction proves to be important.

To apply our framework to our case study on the impacts of SLR, we couple models of SLR with dynamic models of human migration to produce a more comprehensive picture of changing population distributions. We implement our framework with spatial estimates of SLR from the NOAA's Digital Coast dataset [189], small area population projections [176], and a recent machine learning (ML) method for modeling human migration [5]. We model migrations made from *flooded* areas and from *unflooded* areas separately by fitting one ML migration model using "business-as-usual" migration data and one climate change driven model with migration data following Hurricanes Katrina and Rita. While this implementation of our framework highlights its modularity and the importance of separating *types* of human migration - the actual results generated by the individual component models have a large degree of uncertainty - therefore the final estimates are also uncertain. As more sophisticated high-resolution population projections are developed, human migration models are improved, and SLR projections are refined, the precision of our results will also improve.

3.3.1 Modeling framework

Conceptual challenges in coupling human migration models

Traditional strategies for modelling human migration do not lend themselves well to describing climate change driven migration. Current state-of-the-art models of human mi-

gration include the family of radiation models [146, 148], gravity models [10, 11], and machine learning models [5]. The problems are as follows:

First, human migration induced by climate change might not follow historic migration patterns. In fact, using one year of county-to-county migration data from the IRS U.S. migration data sets, Simini et al. [146] showed that a fixed proportion (3%) of the population of a U.S. county will migrate under normal circumstances. This will not hold under SLR, for example, as the entire population in flooded areas will have to move or adapt in other ways. Importantly, in addition to direct inundation due to SLR, climate migrants will be forced to move as climate change effects become more pronounced, directly through the exposure to "high-magnitude events" such as large scale flooding from hurricanes, or indirectly through the "cumulative contribution of ongoing localized events across regions" [190]. The dynamics of these environmentally induced migrations will not necessarily follow those of previously observed migrations. We expect that as social scientists gather more data and knowledge, more refined and informed models of human migration will emerge.

Second, the spatial resolution of each model must be carefully considered. Climate migrants will not necessarily move large distances as they adapt to changing conditions in inundated areas. Indeed, per the US IRS migration data [191], most migrations are made to nearby locations. This phenomenon can be seen in the migrations following Hurricane Katrina in the US, for example, where a majority of destinations were within Louisiana [192]. Hence, it is important to be able to model such nearby migrations, including migrating from the climate affected part to the unaffected part within the same zone, as well as to exclude migrations to areas that are uninhabitable (e.g. inundated by SLR). To do finer scale modeling, the scale of population projections also matters. Population projections are an important input in modeling human migration, and integrating model results over various future population scenarios is crucial for determining uncertainty. However, most existing long term population projections are only done at country or region level scales, e.g. the

variety of population projections defined in the Shared Socioeconomic Pathways [193], and can not be appropriately applied to the finer-scale resolutions at which SLR and migration models will operate.

Third, migration models and population projections are inherently coupled through a feedback loop whereby the population projections at a given time should account for the cumulative effects of climate change driven migration before then. For eaxample, SLR will not happen instantaneously. The SLR influenced population distribution of a location will diverge from that of a "business as usual" scenario as the indirect effects of SLR compound; as more climate migrants settle inland, they will change the migrations patterns of future migrants and so on. However, current models used for population projections at best account for the *direct* effects of SLR where projected populations are made with respect to potentially flooded land [176], however the *indirect* effects must also be considered.

General modeling framework

For some time, t, consider climate change features, $\mathbf{x}^t$, a list of n spatial zones, $\boldsymbol{\theta}^t = [\theta_1^t, \ldots, \theta_n^t]$, which includes a spatial definition and features associated with each zone. Using this information, we want to compute a *migration matrix* $\boldsymbol{T}^t$, where an entry T_{ij}^t represents the number of migrants that travel from zone i to j at time t under a given climate impact model. We propose a general modeling framework for handling this problem which consists of two modules, shown in Fig 3.2: a `CLIMATE` module and a `MIGRATION` module.

`CLIMATE` module. This module uses a "climate impacts model", $CLIMATE(\boldsymbol{\theta}, \mathbf{x})$ to partition each input zone into two new zones: the affected portion and the unaffected portion. Using the best available data, this module should also divide the *features* from the original zone (θ_i) between the affected-portion zone (θ_i^A) and the unaffected-portion zone (θ_i^U). For example, if we have high-resolution spatial population data, then we can split population between the two partitions based on the spatial extent of flooding given a SLR

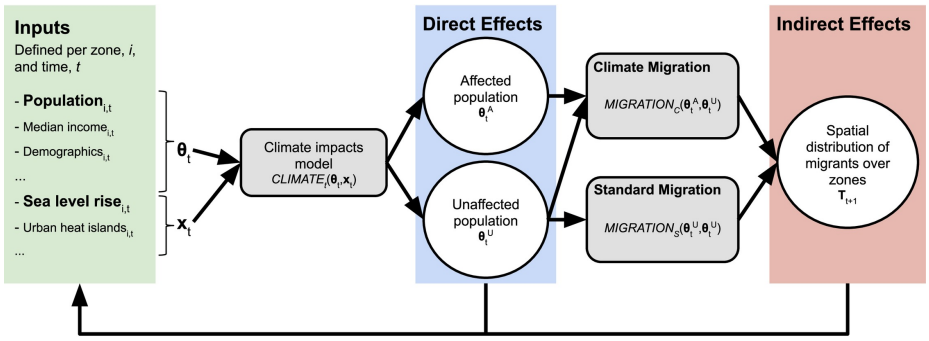


Figure 3.2: **Joint climate change impact and human migration modeling process.** The joint model takes a list of spatial zones $\boldsymbol{\theta}$ and climate change features $\mathbf{x}^t$ as input, and outputs a *migration matrix* $\boldsymbol{T}^t$, where an entry T_{ij}^t represents the number of migrants that travel from zone i to j at time t.

model.

`MIGRATION` module. This module calculates $\boldsymbol{T}$ using the two sets of zones from the `CLIMATE` module. Specifically, this module uses two migration models: 1.) a model for migrations from affected zones $\boldsymbol{\theta}^A$ to unaffected zones $\boldsymbol{\theta}^U$ with the function $MIGRATION_C(\boldsymbol{\theta}^A, \boldsymbol{\theta}^U) = \mathbf{T}'$, where migration is a forced process driven by climate change; and 2.) a model of migrations from unaffected zones to unaffected zones with the function $MIGRATION_S(\boldsymbol{\theta}^U, \boldsymbol{\theta}^U) = \mathbf{T}''$, where migration happens as usual. Finally, this module should aggregate migrant flows from the two migration functions, $\mathbf{T} = \mathbf{T}' + \mathbf{T}''$.

By separating `CLIMATE` driven migration from "business as usual" migrations, our framework forces these dynamics to be considered independently, explicitly bringing up the issue from the first conceptual challenge mentioned in the previous section. Implementations of our framework can use different models for these dynamics if available, or, if such models are not available, fall back to using a simpler model where the simplifying assumption is clear. Our framework also addresses the second conceptual challenge by restricting destinations to only zones marked as unaffected, and allowing the affected population of a zone to choose the unaffected part as its destination. Furthermore, by separating the functionality of the `CLIMATE` module from that of the `MIGRATION` module, the framework allows ablation studies to measure how much results depend on the specific behaviors of

each. The third conceptual challenge revolves around how `CLIMATE` and `MIGRATION` are both temporal processes that form a feedback loop (e.g. SLR will affect migration decisions, which will in turn affect how many people are affected by further SLR, etc.). It is captured in Fig. 1 by including a conceptual link from the estimated migration matrix back into the inputs such as population projections at time $t + 1$.

Finally, our framework provides a methodology by which to calculate the *direct* and *indirect* effects of the climate change impacts model. The *direct* effects are simply the areas that are explicitly affected by the climate impacts model. The *indirect* effects, however, are a function of the changing migration patterns. Formally, a zone is marked as indirectly affected if the difference between the number of incoming migrants in the `CLIMATE` scenario and the baseline scenario is greater than a percentage $d\%$ of the population of that zone. By varying d we can see different intensities of indirect effects.

Our framework instantiated with SLR-based climate impact model and ML-based migration models

We implement our proposed framework considering SLR as the driver of the climate impacts module. An implementation of our framework requires us to define the manner in which a `CLIMATE` function splits features associated with the zones that are affected by climate change, and two `MIGRATION` functions. All three of these steps are discussed in the next two sections. We use SLR spatial estimates based on the NOAA's Digital Coast dataset [189], spatial population projections following the methodology in [176], and a recent machine learning (ML) approach for modeling human migration [5]. All the data sources we use are listed in the S2 Table.

SLR-based climate impacts model First, we use the spatial estimates of SLR from the Digital Coast dataset [189] to determine which census block groups will be flooded under a given amount of SLR, x. The Digital Coast spatial estimates take tidal variability, hydro-

connectivity, probable flooding, and federal leveed areas into account to present a plausible estimate of inundated areas at different *amounts* of SLR, however it is not directly linked to climate projections and does not use more complicated physical flood models. The Digital Coast dataset represents a more serious modeling effort than a "bathtub model" (where using elevation data one marks land under x meters as flooded assuming that sea level will rise uniformly across all coastlines), but can be improved as country-wide sea level rise estimates are improved. We estimate *when* different amounts of SLR considered in the Digital Coast dataset will happen using *climate projections* for two SLR scenarios: **medium** scenario, where 0.9m (3ft) of SLR is experienced by 2100, and **high** scenario, where 1.8m (6ft) of SLR is experienced by 2100. These two SLR projection scenarios are proposed in Hauer 2016 [176], based on methods from the US National Climate Assessment. The medium SLR scenario expects SLR to reach the 0.3m, 0.6m, and 0.9m thresholds in the years 2055, 2080, and 2100 respectively, while the high scenario reaches 0.3m, 0.6m, 0.9m, 1.2m, 1.5m, and 1.8m in the years 2042, 2059, 2071, 2082, 2091, and 2100 respectively.

We pair the spatial flooding estimates at each SLR amount with population projections also following the methodology in Hauer et al., 2016 [176]. Here we create population projections for *every* Census block group in the US ($n = 216,330$). Now, for every 0.3m increment from 0 to 1.8m (corresponding to different years in medium and high scenarios) we have population estimates and area of flooding estimates for every census block group in the US.

With this data, we define the `CLIMATE` module with the $SLR(\theta_i, x_t)$ function that takes a *county*, θ_i, and amount of SLR under either the medium or high SLR scenario, x_t, as input. A given county corresponds to a set of census block groups. A given SLR amount, under either scenario, corresponds to population projection for each census block group, including an estimate of the number of people *affected* by flooding in each block group. For the purposes of modeling migration, we split the county into unaffected and affected portions θ_i^U and θ_i^A respectively. Here, θ_i^U is a "new" county equivalent zone

with a population equal to the sum of the unaffected populations over all the associated block groups. Similarly, θ_i^A, represents the inundated portion of the original county, and will contain a population equal to the sum of affected populations in the associated block groups. We can run $SLR(\theta_i, x_t)$ for all i to get the $\boldsymbol{\theta}^U$ and $\boldsymbol{\theta}^A$ sets. In our case strudy, population is the only feature required for each zone by the `MIGRATION` module, however a similar splitting technique could be used for other block group level features if they can be reliably projected into future scenarios.

Human migration modeling We model human migration between counties in the USA with a recently proposed artificial neural network (ANN) based method [5] that is fit with historic county-to-county migration data from the IRS [191]. This method is similar in functionality to traditional models of human mobility and migration, such as the radiation or gravity models [146, 194, 10]. These models all predict the probability of migration between an origin and a destination as a function of population and distance. The basic intuition behind the traditional models is that probability of migration will be larger with large origin and destination populations, however will decrease with larger distances between the origin and destination. On the other hand, our ANN modeling approach is purely data driven and models the relationship between the probability of migration as a highly parameterized nonlinear function of the features of origins and destinations.

More specifically, our ANN models estimate P_{ij}, the probability that a migrant which leaves an origin county, i, will travel to a destination county, j, as a non-linear function of: origin population, m_i, destination population, m_j, great-circle distance between the two, d_{ij}, and the "intervening opprtunities" between the two, s_{ij} (this is the total population in the circle centered at i with radius d_{ij}, not including m_i or m_j). These features are the same features used by traditional radiation and gravity models, and depend solely on population and distance. While the ANN uses the same set of features, it does not have a fixed functional form and can learn complex non-linear relationships between the features

and the target output (P_{ij}).

To compute T_{ij}, the number of migrants that travel from i to j, we need to know the number of migrants that are attempting to leave i. If we say that the number of migrants leaving zone i is of the form $g(m_i) = \alpha m_i$, where α is some coefficient that specifies the fraction of the total population that will migrate, then $T_{ij} = g(m_i)P_{ij}$. This function g is called the production function. Now we can define the `MIGRATION` functions, $MIGRATION_C$ and $MIGRATION_S$, which represent the climate migrants and "business as usual" migrants respectively, by training two instances of our ANN model, and forming respective production functions $g_C(m_i)$ and $g_S(m_i)$ by choosing α_C and α_S.

We fit the $MIGRATION_C$ model by finding hurricane affected counties from the IRS migration data from 2004-2011 and 2011-2014. Specifically, we search for migration data points (i.e. pairs of counties) in which the origin county was a coastal county that observed an over 100% *increase* in outgoing migrations with over 1,000 total outgoing migrations[3]. This "filter" highlights counties that have potentially been affected by hurricanes or other natural disasters and indeed finds seven counties from 2005 that were heavily impacted by hurricanes Katrina and Rita: St Bernard, Orleans, Cameron, Plaquemines, Hancock, Jefferson, and Harrison (which matches literature estimates of the most damaged counties [192]), as well as Liberty County, GA from 2006. The only historical explanation we can find for a sudden increase in outgoing migration in Liberty County is the deactivation of a large US military division stationed within the county that year. Considering this, we fit our $MIGRATION_C$ model using the data from the seven counties that were most seriously affected by hurricanes Katrina and Rita. As background, Hurricane Katrina struck the city of New Orleans and the broader Louisiana and Mississippi coastline on August 29, 2005 causing widespread flooding and wind damage. Less than a month later, on September 25, Hurricane Rita also struck the Louisiana coast, exacerbating damage in New Orleans and causing extensive damage to counties in the western portions of the state. Over 1500 peo-

[3]The reporting methodology in the IRS migration dataset changed between the 2010-2011 data and 2011-2012 data, therefore we cannot measure percent increase in outgoing migrations between them.

ple were killed and over 80% of the city of New Orleans was flooded as a result of these hurricanes. Followup studies and Census estimates showed that New Orleans only contained around half of its pre-hurricane population within a year of the storms. By training our ANN with these counties we allow the model to pick up on the dynamics of migrations after extreme flooding events.

We train an ANN, $MIGRATION_C(\theta_i^A, \theta_j^U) = P'_{ij}$, using all 7×3099 pairs of counties from the 2005-2006 IRS data that include one of the seven previously mentioned *affected* counties as an origin and an *unaffected* county as a destination. Similarly, we fit another ANN using the rest of the IRS migration data, $MIGRATION_S(\theta_i^U, \theta_j^U) = P''_{ij}$. Due to our assumption that all people in flooded areas will have to migrate, the production function for climate migrants is given as the identity, $g_C(m_i) = m_i$. This forces the entire population of the affected portions of counties to become migrants. For "business as usual" migrants, we use the production function, $g_S(m_i) = 0.03m_i$, due to the observation by Simini et al. [146] that 3% of a county's population will migrate under normal conditions each year. Given these: $T'_{ij} = g_C(m_i)MIGRATION_C(\theta_i^A, \theta_j^U)$, and $T''_{ij} = g_S(m_i)MIGRATION_S(\theta_i^U, \theta_j^U)$. With these definitions we can build $\mathbf{T}'$ and $\mathbf{T}''$ by running the climate migration ANN and "business as usual" migration ANN for all pairs of counties.

In Section 2 of the S1 Appendix we show a similar implementation using the Extended Radiation model [148] to implement the `MIGRATION` functions. In Section 3 of the S1 Appendix we describe our ML model and give validation results comparing our ANN based migration model to other human migration models on the task of predicting inter-county migrations in the US.

3.3.2 Results

We categorize the effects of SLR into two types: *direct effects*, which are a direct consequence of SLR, and *indirect effects*, which are a consequence of changing migration

patterns due to SLR. We present the spatial distribution and magnitude of these effects in Figs 3.3 and 3.4. People that live on flooded land who will have to move away are accounted for in the *direct effects* of SLR. People that live in counties that experience a larger number of incoming migrants in the flooding scenario relative to the baseline scenario with no SLR are accounted for in the *indirect effects* of SLR.

In Fig 3.3 we show the spatial distribution of changes in migration patterns. In the top panel, counties experiencing any flooding (i.e. that are directly affected by SLR) are highlighted in blue, while the remaining counties are colored according to how many additional migrants they receive in the 1.8m SLR scenario. The bottom two panels of Fig 3.3 show the difference in the number of incoming migrants between the SLR scenario and the baseline scenario for incoming migrants from *unaffected* counties and *affected* counties. The top panel is the sum of these two maps, and shows this difference for incoming migrants from *all* counties. From these maps we can see that the primary destination of climate migrants are counties just inland of their origin, but climate migrants also move farther towards large cities that offer more opportunities.

In Fig 3.4 we show the magnitude of the *direct* and *indirect* effects as well as the spatial distribution of the indirectly affected counties. We calculate whether a county (and hence its population) has been *indirectly affected* at a level $d\%$ through the methodology described in the General modeling framework section. On average, in business as usual conditions, 3% of a county's population migrates each year [146]. Thus, if $d = 3\%$, we would mark a county as indirectly affected if we observe twice as much migration into that county than the average migration rate of the US. We assume that as d increases, the effects will be stronger as there will be a significant strain on the resources in that particular county.

The graph in the bottom panel of Fig 3.4 shows the direct and indirect effects of SLR in terms of number of people affected for amounts of SLR in the range from 0.3m to 1.8m in 0.3m increments. The map in the top panel shows which counties in the United States are indirectly affected at different threshold values of d. In both plots the indirect effects

are shown for five different values of d: 0.5% 1%, 3%, 6%, and 9%.

We can see that the *indirect* impacts of SLR grow at much faster rate than the *direct* impacts (Fig 3.4). In the high SLR scenario by the year 2100 there are $\approx$ 13 million people *directly* affected, in $\approx$ 50 thousand km^2 of flooded land, however there are almost twice as many, $\approx$ 25 million people, *indirectly* affected at the 9% threshold due to changing migration patterns and magnitudes. This d=9% threshold indicates that these people live in areas which will experience four times as many migrants as they would compared to a baseline scenario. Even under the moderate assumption of 0.9m SLR by 2100 there will be 24 million people that live in counties considered indirectly affected at a d=3% threshold. Under the same threshold with a SLR of 1.8m by 2100, there will be 120 million people, over $\approx 1/3$ of the population of US, living in counties that will see a doubling in the number of annual incoming migrants.

The map in Fig 3.4 shows that these *indirect* effects relative to county population will be distributed unevenly over the US. Most effects are seen in the Eastern US, where there are more vulnerable coastal populations. Of particular note are southern Mississippi and southeastern Georgia, where large groups of counties are estimated to see indirect effects in the $> 9\%$ category. The Midwest is also projected to see large indirect effects, even though the magnitudes of incoming migrants are smaller than counties closer to the coast. This can be explained by the relatively small populations and baseline levels of incoming migrants. The greater magnitudes of migrations from higher population areas causes *some* migrants to select these midwestern areas as destination, which could cause disproportionally larger *indirect* effects.

In general, we find that previously "unpopular" migrant destinations (areas with relatively low numbers of incoming migrants) would be more popular solely due to their close proximity to counties that experience "direct effects". The East Coast will experience larger effects than the West coast because of the large coastal population centers and shallower coastlines, indeed, all counties adjacent to coastal counties on the East coast are marked as

indirectly affected. Existing urban areas will receive the largest magnitudes of migrants, as they represent the most attractive destinations, which will accelerate the existing trends of urbanization. We find that the southeast portion of the United States will experience disproportionately high effects from SLR-driven flooding due to the large vulnerable populations in New Orleans and Miami. These results show that by driving human migration, the impacts of SLR have the potential to be much farther reaching than the coastal areas which they will flood.

Finally, we examine the effects that the choice of migration model has on our results in S1 Appendix. S1 Fig. and S2 Fig. show results in the same format as Figures 2 and 3, however with an implementation of our general framework that uses the extended radiation model of human migration. Further discussion can be found in S1 Appendix.

3.3.3 Discussion

The framework we propose here (Fig 3.2) introduces a systemic approach to couple models of climate change impacts with models of human migration. The framework has two improvements over our understanding about of the impacts of climate change on population by closely considering the corresponding effects on migration patterns. First, we posit that to capture the impacts of climate change, we need to consider how climate change affects population directly, "direct effects," and indirectly through migration-driven heightened pressures on population centers, "indirect effects." Second, we argue that calibrated models of climate impacts need to account for the fact that human responses to climate change, migration in our particular case, will differ from business as usual responses.

We apply our framework to analyze the impacts of sea level rise on human migration patterns and levels. We couple spatial estimates of SLR with human migration models and show that the effect of SLR on human populations could be more pervasive and widespread than anticipated, with almost all counties in the US receiving some number of additional migrants due to SLR induced flooding. Our results are the first step in understanding the

socio-economic impacts of climate driven migration and more research is needed to understand how increased migration affects populations across different destinations.

While our framework is flexible enough in theory, in practice it will require assumptions specific to each application. These assumptions can be the result of knowledge gaps or data limitations. In our application to SLR, we have made several such assumptions that affect how we interpret the results of the coupled SLR/migration models. For example, we have assumed that people move due to SLR only when their homes are flooded. It is of course possible, that people will move because their business or jobs are affected. According to the 'Nuisance Hypothesis' from Keenan et al. [195] housing prices are affected by people's perceptions as to whether or not a property is at risk of flooding. This could impact "pull" factors of migration. While we cannot consider this channel in our current application due to data limitations, our framework allows to expand in this direction once more data is available.

Because we rely on machine learning techniques, we forfeit the explanatory power of our migration model in favor of a more accurate prediction. This limits our capacity to derive specific policy recommendations, but this approach is suitable for the purposes of our research question as it conceives a much more flexible methodology to analyze future migration. This black-box approach can be further calibrated as more data on similar temporal and spatial scales for empirical studies to explain migration behaviors becomes available [180].

The limitations of our application notwithstanding, we find some of our results are similar to previously published estimates of human migration under SLR in the USA. For example, like in Hauer 2017 [187], inland areas immediately adjacent to the coast, and urban areas in the southeast US will observe the largest effects from SLR driven migration. Our framework for modeling human migration, however, reveals several notable differences. For example, compared to Hauer 2017 [187], our results show more incoming migrants to Houston and Dallas - two larger cities closer to affected coastal areas. This result follows

from our framing requirement that flows predicted under climate change do not necessarily follow historical flows. According to Hauer 2017, the Austin, Texas Core Based Statistical Area is expected to observe the largest effects out of all destinations, with over 800 thousand incoming migrants due to SLR. This result follows because Austin has consistently been one of the fastest growing US cities over the past decade, which is captured and projected by a time series based migration model. Our migration model instead captures the dynamics of human migration between US counties based on population and distance features, and uses this to predict flows between counties without regard to potential short term historic trends. Thus, our framework has the benefit of predicting flows between pairs of counties for which there are no historical flows. Our application also shows the "indirect impacts" could be important in magnitude. We observe that migrants from unaffected areas, that would previously move to coastal areas, will relocate to larger population centers. The counties surrounding Los Angeles in particular could see tens of thousands of migrants that are not coming from affected areas, but must choose a different location because their preferred coastal destination are now flooded.

While our application results are limited, our framework has shown conceptually how to think about widespread impacts of climate change. Moreover, as various aspects of human migration are better understood, especially ones related to environmental pressures, better models of human migration are created, and SLR flooding estimates are improved - with finer resolution population projections, uncertainty estimates, and models of the potential spatial effects of SLR such as expected flood frequency - we will be able to update our framework to produce more refined and comprehensive results.

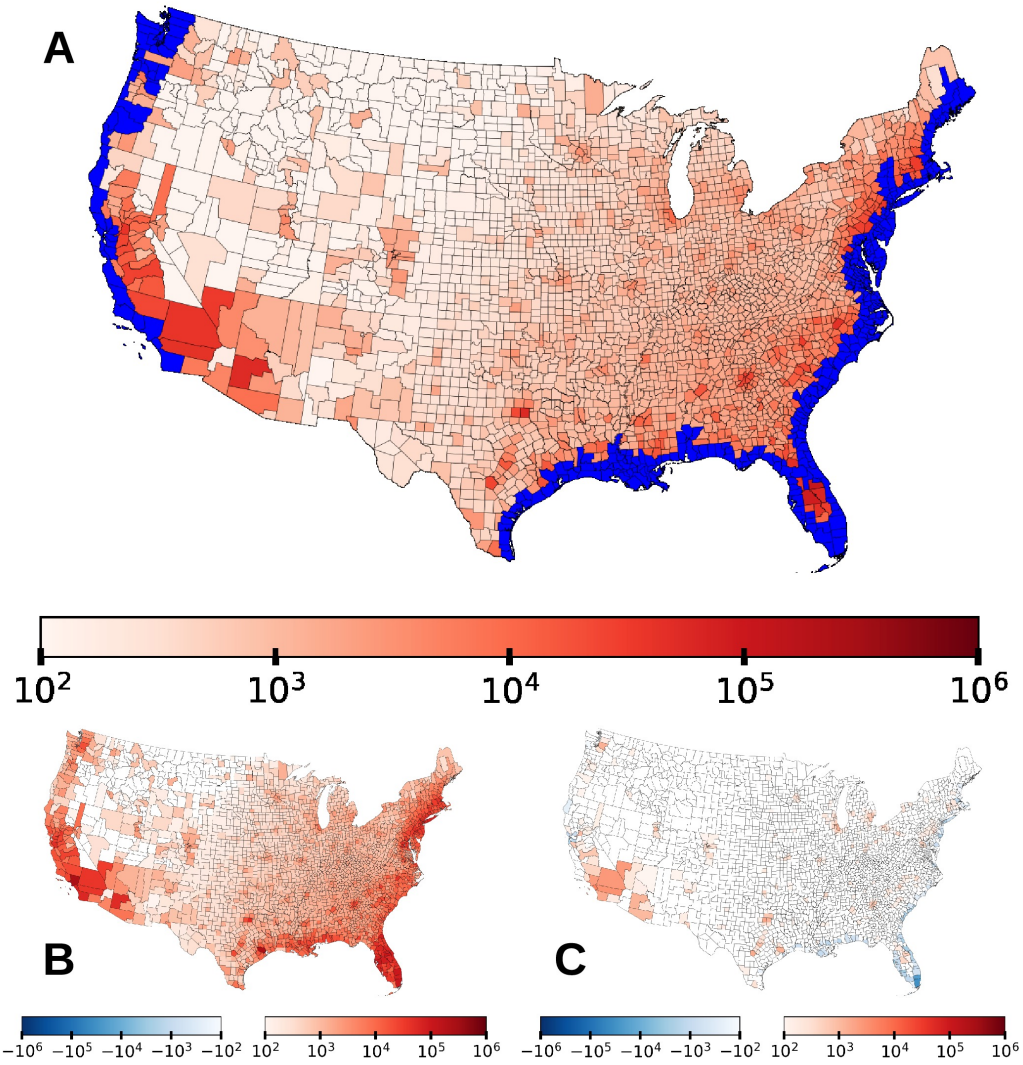


Figure 3.3: **Spatial distribution of the direct and indirect effects of SLR on human migration.** The **top** panel shows all counties that experience flooding under 1.8m of SLR by 2100 in blue and colors the remaining counties based on the number of additional incoming migrants per county that there are in the SLR scenario over the baseline. The **bottom left** map shows the number of additional incoming migrants per county in the SLR scenario from only flooded counties. The **bottom right** map shows the number of additional incoming migrants per county in the SLR scenario from only unflooded counties. Color gradients are implemented in a log scale.

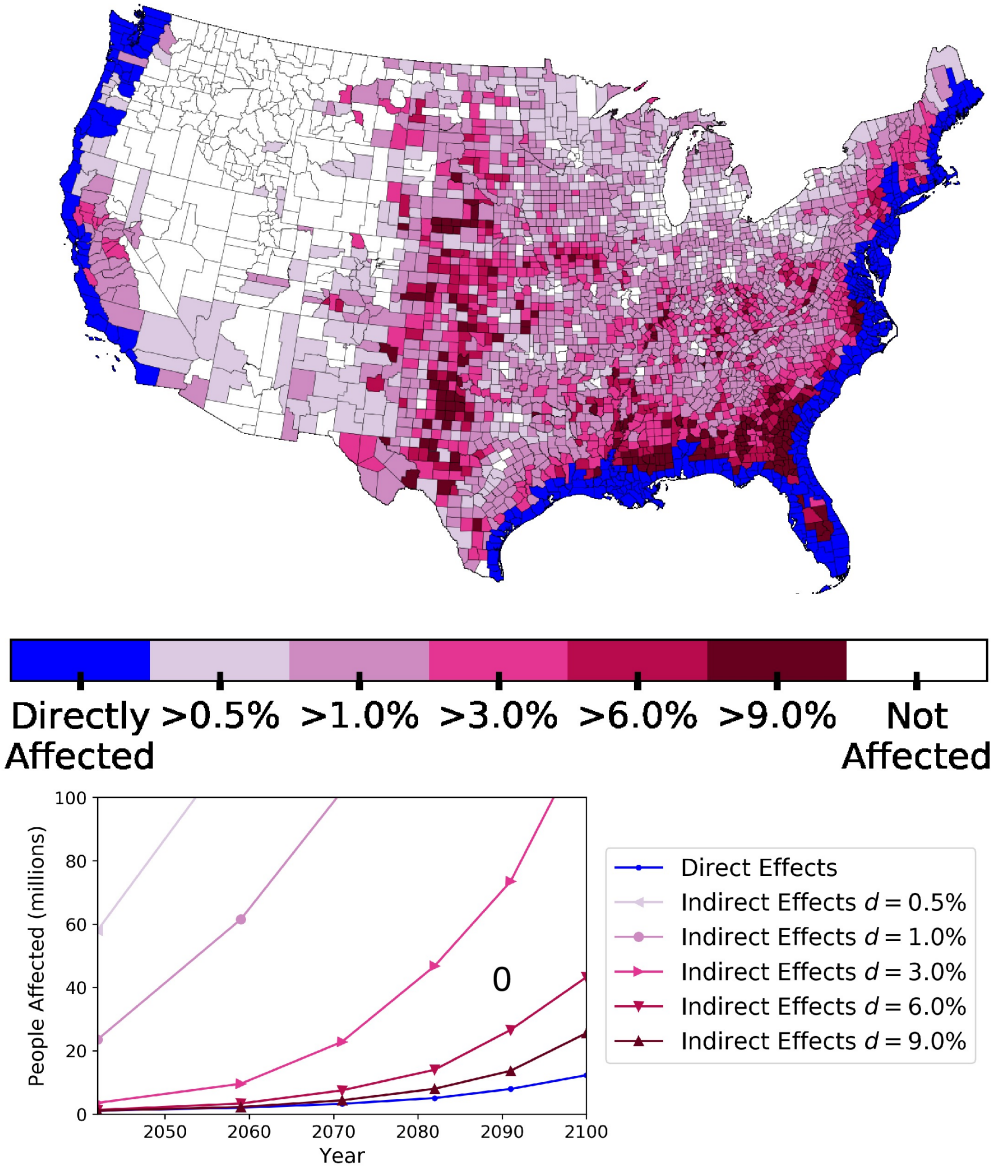


Figure 3.4: **Impacts of SLR due to flooding and human migration for a range of SLR scenarios.** We say that a county is indirectly affected by SLR if the difference between the number of incoming migrants to the county in the SLR scenario and the number of incoming migrants in the baseline scenario, i.e. the number of *extra* migrants in the SLR scenario, is greater than some percentage, d, of that county's population. In the **top** panel we show the spatial distribution of counties that are considered indirectly affected at different threshold values of d for the 1.8m SLR case in the southeast portion of the United States. In the **bottom** panel we show the number of people that are directly and indirectly affected under the same threshold values of d for the entire United States. For both plots we show aggregate impacts for five different values of d: 0.5%, 1%, 3%, 6%, and 9%.

CHAPTER 4
CONCLUSION

In this dissertation we have developed machine learning methodology that can be used to facilitate various decision making endeavors that rely on geospatial data over large scales and applied them in various settings.

In Chapter 2, with deep learning techniques and remote sensing data, we show how new methods for training supervised models with different low resolution data sources, and with humans in athe loop can be used to improve land cover mapping efforts. We show how new methods for unsupervised training of models can lead to better feature extractors that are able to be fine-tuned on geospatial learning problems (including land cover mapping) with limited sized training data sets. We also show how training data over polygons can be used to learn to aggregate predictions made at a raster level in the context of human population estimation from satellite imagery. These methods allow us to tackle the problems of land cover mapping and human population estimation in different real world settings, i.e. with varying amounts of input and labeled data.

In Chapter 3, with machine learning methods and human migration data, we have shown how a new loss function can be used to train migration models that outperform existing baselines. We use these migration models coupled with sea level rise models in a general framework to reason about the direct and indirect effects of sea level rise. We instantiate this framework in the US and create population projections for future sea level rise scenarios.

The widespread collection of geospatial data provides us a huge opportunity to measure vital quantities about the earth and human society over time. We need to continue to develop methods that can exploit this data and use it to advance sustainable development goals.

Appendices

APPENDIX A
MACHINE LEARNING APPROACHES FOR ESTIMATING BUILDING ENERGY CONSUMPTION

A.1 Introduction

There is substantial evidence to suggest that different configurations of the built environment are closely associated with variations in energy consumption and climate altering greenhouse gas emissions [196, 197, 198, 199, 200]. While the relationship between urban form and energy use in transportation has been well studied, we know far less about the impact of urban form on residential and commercial energy demands [201, 202, 203, 204]. A 2009 study commissioned by the National Academy concluded that increasing development densities leads to modest savings in energy use in transportation, and by extension, a reduction in greenhouse gas emissions [205]. Yet, if our interest is in building energy efficient communities, a more comprehensive set of attributes of the built environment need to be examined to determine whether increasing development densities actually lead to energy savings. The estimation of building energy consumption at the scale of small urban areas is difficult without building level data and few studies have attempted to provide energy footprints for residential and commercial buildings at neighborhood scales. This paper fills some of that gap by providing a generic technique for estimating commercial building energy from publicly available data in the U.S.

According to the 2015 annual energy consumption data released by the U.S. Energy Information Administration (EIA), residential and commercial buildings consumed 39 quadrillion Btus., representing 40% of total energy consumption in the United States [206]. Similarly, according to the European Commission [207], building energy consumption accounts for 40% of the total energy consumption in the EU. Globally, the building sector accounted

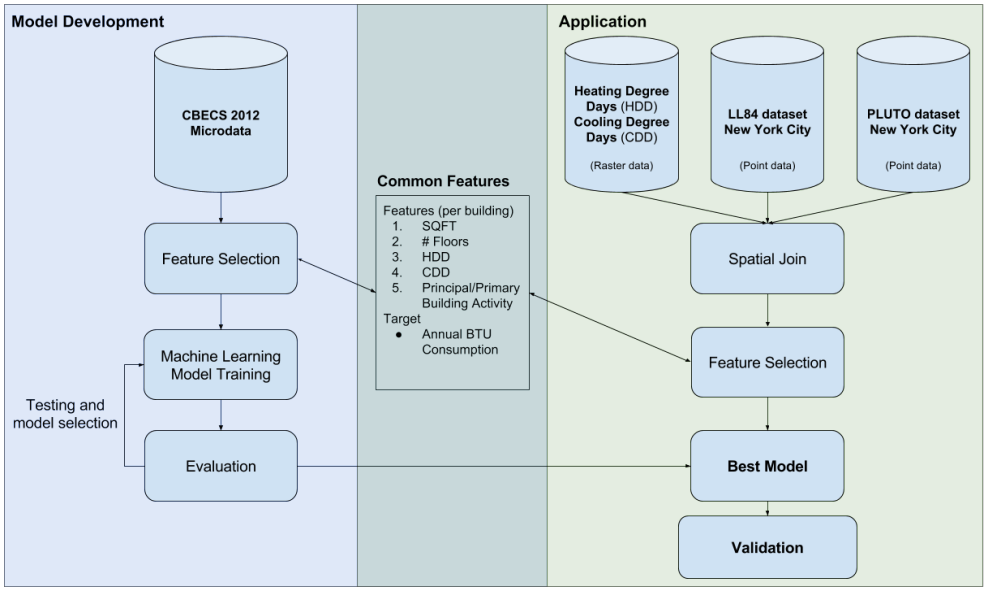


Figure A.1: Modeling framework.

for approximately 32% of energy consumption in 2010 [208]. Thus, advanced economies spend a particularly large percentage of their energy in buildings compared to developing countries. While the EIA releases highly detailed annual energy consumption estimates by sector for the U.S. as a whole, it is useful for local policy makers to have small area or neighborhood level estimates of energy consumption. Without access to fine scale data on energy use, urban planners will not be able to benchmark the effects of environmental or climate related policies affecting different sections of the urban region or make confident predictions about the outcomes of proposed policies. Machine learning models can also help city and regional planners predict the energy burdens that could result from alternative urban growth patterns and global warming scenarios. Spatial energy consumption information at a granular scale is therefore crucial to fulfilling sustainability goals.

One way of estimating building energy consumption, in the absence of actual sensor data, is to create physical building models with a "template" of representative buildings, then run thermodynamic simulations to estimate the energy demands [209]. These "engineering" models of building energy consumption are computationally expensive and cannot capture the wide variety of different buildings present in cities, as modeling each type of

building requires very detailed input data, which is costly to collect. Statistical models can be used to fill the gaps where resources are too limited to use physical models, or the scale of the study area makes physical modeling impractical.

We aim to model commercial building energy consumption at the building level using machine learning models. This statistical approach avoids expensive physical modeling efforts, and is able to provide reasonable estimates that can be validated against existing building level energy consumption databases. Specifically, we train machine learning models on the 2012 Commercial Building Energy Consumption Survey microdata [210], then validate this approach using the Local Law 84 (LL84) dataset from New York City. We show how our models can be used to create comprehensive metropolitan wide commercial energy consumption maps by applying them to 73,388 commercial buildings in Atlanta, GA. These maps will help city planners better understand the relationships between urban form and energy consumption, and plan for the future. Our models purposefully only rely on a limited set of building level features, namely: square footage, principal building activity, number of floors, and heating and cooling degree days, so that they can be applied to *any* metropolitan area in the United States. Furthermore, to facilitate the wider adoption of our methods in other metropolitan areas throughout the US, we have released the code and trained models used in this paper in a public GitHub repository[1]. The code and instructions provided in the GitHub repository can be used to reproduce the modeling and validation results from this paper, and to apply trained models in new settings. In general, the machine learning modeling approach for broad commercial building energy consumption prediction presented in this work is a novel step toward better understanding the energy consumption landscape in the United States.

[1]The code and trained models are available at: `https://github.com/SEI-ENERGY/Commercial-Energy/`

Table A.1: Commonly used abbreviations.

Abbreviation	Description
CBECS	Commercial Building Energy Consumption Survey - A national (for the US), statistically representative survey of commercial building characteristics and energy consumption published by the Energy Information Administration.
LL84	New York City Local Law 84 data - A dataset published yearly by New York City on the energy consumption of commercial buildings.
MFBTU	Major fuel consumption in BTUs - This is the total energy consumption of a building, i.e what our models predict.
PBA	Principal building activity - The main use category of a building, e.g. “Office”, “Education”, “Nonrefridgerated warehouse”.
TAZ	Traffic analysis zone - The name given to small geographic areas used for planning in many cities. We spatially aggregate our model’s commercial building energy consumption estimations in Atlanta using these zones.
HDD	Heating degree days - The total number of degrees below 65 Fahrenheit summed for each average daily temperature in a year. This a proxy measure for the total energy needed to heat a building in a year.
CDD	Cooling degree days - The total number of degrees above 65 Fahrenheit summed for each average daily temperature in a year. This a proxy measure for the total energy needed to cool a building in a year.

A.2 Related work

Methods for predicting building energy consumption can be categorized into three groups: engineering methods (white-box models), statistical methods (black-box models), and hybrid approaches (grey-box models) [209, 211, 212]. Engineering methods physically model building energy consumption by simulating the laws of thermodynamics using extensive building level data. This method cannot be applied precisely at the urban scale, due to its large data and computational demands, however it is used to estimate the energy consumption of a small typology of buildings which are then aggregated over entire urban areas [213, 211, 212]. Statistical methods for estimating building energy consumption aim

to directly regress energy consumption values on associated building and climate variables. In general, machine learning methods (such as the methods used in this study) fall into this category, although Zhao and Magoulès have separated machine learning based studies from linear regression model based studies in their review [209]. Hybrid methods involve a combination of both engineering and statistical models, and use the output from engineering models as an input to statistical models. The purpose of these models is to offset some of the constraints involved with physical modeling (such as the inability to model every building in a district) with the flexibility of statistical approaches [214].

A commonality between the engineering, statistical, and hybrid methods is that they are all limited by the availability of relevant data. Indeed, availability of data is crucial for any statistical modeling approach, but our method lowers the bar for data requirement as discussed later. Mathew et al. discuss big data applications of the US Department of Energy's building performance database (BDP) [215]. The BDP has data for both residential and commercial buildings on a larger scale than either the Commercial Building Energy Consumption Survey (CBECS) or the Residential Building Energy Consumption Survey (RECS), however can only be used in benchmarking applications. Access to fine grained data, such as that collected by BDP, will be crucial for development of more accurate and relevant statistical based models [216].

Linear statistical models have been used in studies for predicting energy consumption at both the building and zone level resolutions. Boulaire et al. use robust linear models to model energy consumption at the zone level in NSW, Australia [217]. Kuusela et al. use a lognormal modeling framework to model electricity consumption from aggregate building features at the zone level of a Finnish city [218]. Kontokosta use robust linear models to estimate building energy consumption of residential and commercial buildings using 2011 New York City's Local Law 84 (LL84) dataset [219]. While these models are easily interpretable, machine learning models are better suited for modeling the complex relationships between building level characteristics and energy consumption since such

models have fewer constraints about the statistical relationships among variables.

Previous studies have shown that machine learning models out-perform linear models in modeling building energy consumption. Tso et al. use linear regression models, decision tree models, and neural networks to model residential electricity consumption at the building level in Hong Kong [220]. The study splits the dataset across the summer and winter seasons and trains models separately for each season.Similarly, Fan, Xiao, and Wang use an ensemble of machine learning models to predict the next-day building energy consumption of the "International Commerce Center" in Hong Kong with good results [221]. Wei et al. use two linear models and four non-parametric machine learning models to estimate gas and electricity consumption at a zone level in London [222]. Similarly, Yalcintas et al. train an artificial neural network and multiple linear regression models to predict the energy use intensity values (kWh per square meter) with the 1999 CBECS data [223]. This study only uses one category of building from the CBECS dataset, and categorizes the target values to convert the problem into an easier classification problem. These three studies all find that machine learning models perform better than linear regression based models, however they are limited both by the few models they consider, and the smaller datasets they use. In our work we consider a broad range of machine learning models, and use as much of the CBECS data as possible with the objective of creating a general model for estimating commercial building energy consumption.

Finally, the most similar work to ours is by Howard et al., which uses the LL84 dataset to estimate robust linear regression models for predicting energy consumption in commercial and residential buildings in New York City [224]. The authors calibrate the linear model using the building function, square footage, and energy use intensity features from the New York specific data, and the final predicted energy consumption is disaggregated into several different end uses based on CBECS and RECS data. The final estimates were aggregated at the block area level to provide spatial energy consumption distributions. Our work is different from this as we train multiple machine learning models on the national

CBECS data, which we then apply to specific metropolitan areas. We are not concerned with the specific energy end-uses in commercial buildings, as this would require more detailed data to model and validate, but instead focus on estimating generally capable models that can be used to create acceptable estimates of total energy consumption using as few features as possible.

A.3 Data and methods

Our two main objectives are to 1.) train machine learning models to predict the annual major fuel, or the combination of electricity, natural gas, and fuel oil, consumption, of commercial buildings from easily accessible descriptive features of buildings, and 2.) validate the models' ability to be applied to specific metropolitan areas. Specifically, we train and test our models using national survey data from CBECS, then use true energy consumption values from New York City's Local Law 84 (LL84) dataset to validate the ability of the nationwide CBECS-trained models to be applied accurately to a specific metropolitan area. In Section A.3.1 we describe the datasets we use in further detail, in Section A.3.2 we explain the details of our methods, and in Section A.4 we present our results and discussion. We give a listing of commonly used abbreviations in Table A.1.A graphical overview of the methodology that we follow in this work is given in Figure A.1.

A.3.1 Data

The CBECS microdata is released by the U.S. Energy Information Administration (EIA) approximately every five years. The latest microdata, from 2012, contains 6,720 rows of data, where each row represents some of the estimated 5.6 million commercial buildings in the U.S. [210]. Specifically, each row contains the features, or information on numerous characteristics, of a particular representative building gathered through the CBECS 'Building Survey' questionnaire. These features include information such as: the square footage of the building, the principal building activity (PBA), heating and cooling degree days,

number of employees, etc.

The New York City Benchmarking Law, known as Local Law 84 (LL84), requires buildings that are over 50,000 square feet, or lots with buildings with over 100,000 square feet combined, to report their annual energy and water consumption values in a standardized manner using the EPA's portfolio manager database [225]. This consumption data, along with some of the building characteristics, have been released annually since 2011. We use the most recent "2016 Energy and Water Data Disclosure" data that contains information on energy consumption from 2015. This dataset has 13,223 rows of data, where each row represents a building or collection of buildings on a single lot, in one of the five boroughs of New York City. Each row of data contains feature such as: total square feet, year built, primary building activity, and energy use intensity (kWh/ft^2).

An important component of building energy demand is the amount of energy used to heat and cool the building. Heating and cooling degree day variables have been shown as useful indicators of this demand [226], and are included in the CBECS dataset, however are absent from the LL84 dataset. We augment each row in the LL84 dataset with heating degree day (HDD) and cooling degree day (CDD) features from 2015 CDD and HDD raster maps. These raster maps were calculated according to the methodology found in [227], from the daily average temperatures predicted by an aggregate of 11 climate models run at Oak Ridge National Laboratory [228]. We further join the LL84 dataset to the New York City PLUTO dataset [229] in order to get more information, such as the number of floors, for each building in the LL84 dataset. After this processing, we have information on 2,612 commercial buildings, which we will simply refer to as the LL84 dataset.

Preprocessing the Commercial Building Energy Consumption Survey (CBECS)

The procedure we use to preprocess the raw 2012 microdata into the format we use in the study is as follows[2]:

[2]For code to reproduce this cleaning process, see the GitHub repository at `https://github.com/SEI-ENERGY/Commercial-Energy`

- Remove rows where the ZMFBTU field is not 2 or 9 (i.e. only keep rows where our target value, MFBTU is present).

- Replace values where NFLOORS $= 994$ with 20. Replace values where NFLOORS $= 995$ with 30. The 994 value represents 15 to 25 floors and the 995 value represent greater than 25 floors. These choices of values will let algorithmic approaches treat the NFLOORS feature as an integer value.

- Discard all features that start with the letter 'Z' (i.e. features that take the form 'Z*****'). These fields report whether or not another feature is: 'reported', 'imputed', 'estimated', 'missing', or 'inapplicable', and will not be useful for our models.

- Discard all of the FINALWT columns.

- Discard features (columns) that have over 25% of missing values, then impute remaining missing values per feature, with the most common value for that feature.

- Perform a one-hot encoding on the PBA feature by removing the original feature, and adding 20 new features, where each feature represents a particular PBA. For each row of data, the new feature that represents the original PBA value will be set to 1, and the remaining will be set to 0.

This process will result in a data table, $\mathbf{X}$, containing 179 features for each of the 5099 buildings, $\mathbf{X} \in \mathbb{R}^{5099 \times 179}$, with a single target vector, Y, representing the MFBTU value for each building, $Y \in \mathbb{R}^{5099}$. This is trimmed down from the original dataset that had 6720 rows and 1119 features. Each building in $\mathbf{X}$ falls into one of 20 different classes according to the buildings principal building activity, or PBA. The numbers of building per PBA can be seen in Figure A.2. The distribution of the MFBTU values in Y can be seen in Figure A.3.

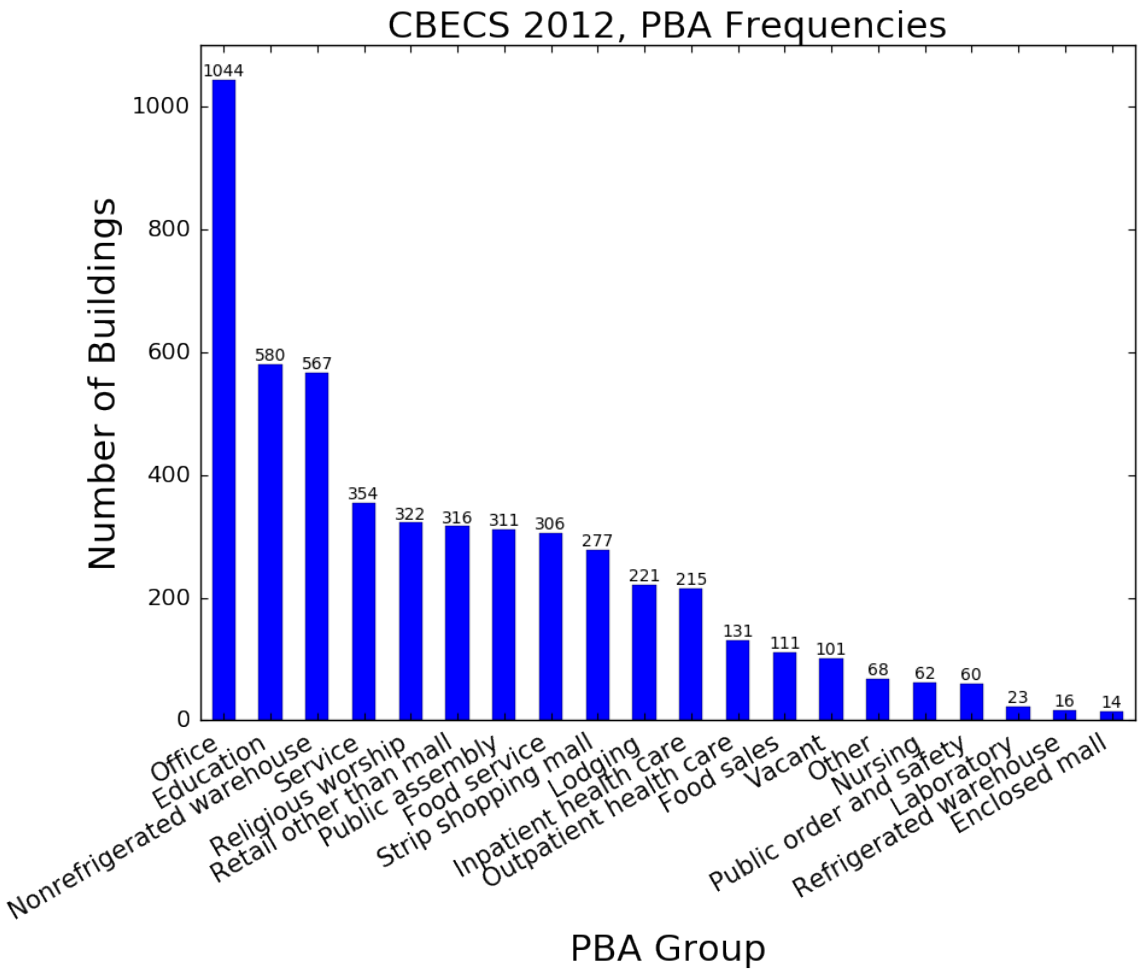


Figure A.2: Number of samples of each class of building (PBA), after preprocessing, in the CBECS dataset.

Preprocessing the Local Law 84 Dataset (LL84)

We join the latest LL84 dataset from the 2015 calendar year with the latest PLUTO 16v2 dataset, and cooling and heating degree day rasters from climate model outputs. This process will enable to construct the same *common feature set* for NYC buildings that we use in the CBECS dataset.

The steps we follow to get the final LL84 dataset used in the study are as follows:

1. Each building/tax lot in New York City has an unique identifying number called the Borough, Block, and Lot (BBL) number. We use the BBL field from both the LL84

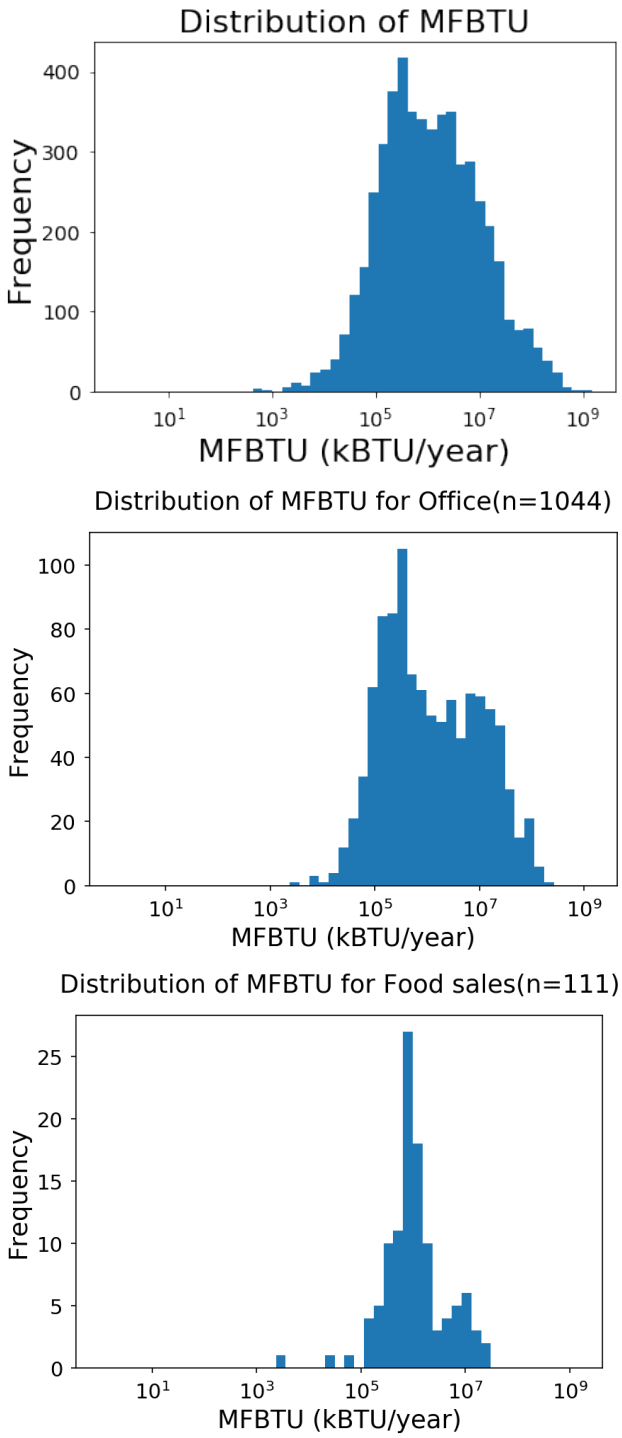


Figure A.3: MFBTU distributions for: all CBECS data (**top**), buildings in the "Office" class (**middle**), and buildings in the "Food Sales" class (**bottom**). The top panel shows how the `MFBTU` target values follow a log-normal distribution, and therefore, how the log transformation of the `MFBTU` values will follow a normal distribution.

and Pluto datasets to perform an inner join on the two datasets.

2. We drop all rows for which any of the following fields are missing: "Primary Property Type - Self Selected", "Site EUI (kBtu/ft^2)", "Property GFA - Self-reported (ft^2)", or "NumFloors".

3. We map the "Primary Property Type - Self Selected" field to CBECS "Principal Building Activity" field according to the custom mapping defined in Table A.2. Any rows with a "Primary Property Type - Self Selected" value in the LL84 field that is not present in Table A.2, such as "Multifamily Housing", are dropped from the dataset. We then perform an one-hot encoding of this mapped `PBA` field using the same method used in the CBECS processing steps.

4. The PLUTO dataset comes with shapefiles that have a polygon for each row (MapPLUTO). We calculate the centroid points for each of these shapes, which lets us associate a latitude/longitude point with each row in the LL84 dataset.

5. We use the latitude/longitude points for each row to lookup the cooling and heating degree day values from rasters derived from an average of 11 climate models run at Oak Ridge National Laboratory. The heating and cooling degree values are calculated from the 2015 average daily temperature as reported by the climate models, with a temperature of 65 Fahrenheit used as the base value for the degree day calculation. The rasters are shown in Figure A.4.

6. For each row, we convert the energy use intensity value, "Site EUI (kBtu/ft^2)", to total energy use by multiplying by the total square footage. These values are then used as the target values in the LL84 dataset.

The result of this process is a set of 2,612 commercial buildings from New York City, each with all features in the *common feature set*: square footage, number of floors, cooling degree days, heating degree days, and PBA, as well as the total energy consumption value.

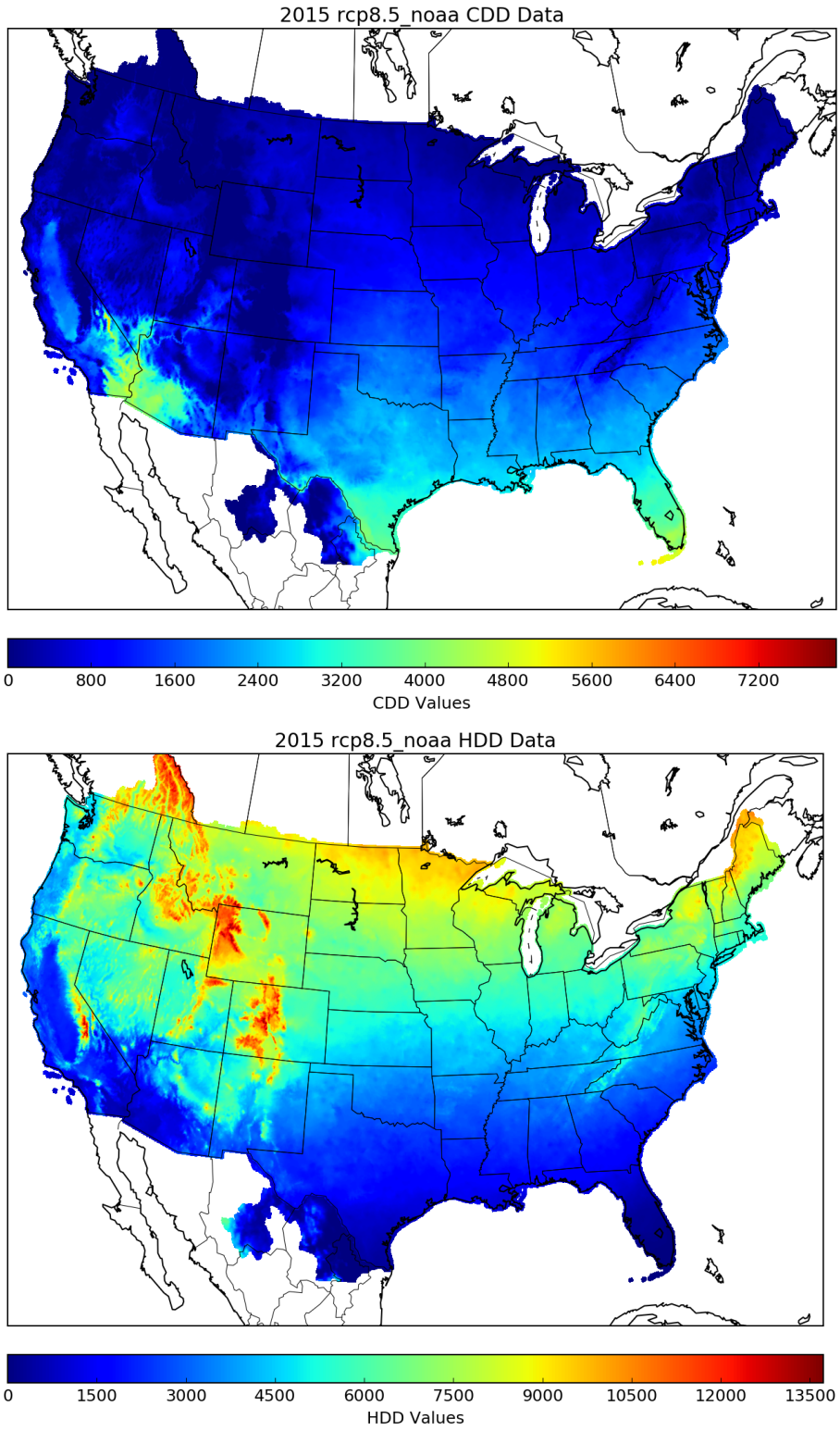


Figure A.4: Cooling and heating degree day rasters for 2015.

A.3.2 Modeling commercial building energy consumption

We want to predict the 'Annual Major Fuel Consumption' (the `MFBTU` field in CBECS) of commercial buildings by only using some features of the buildings. We express this objective in a machine learning regression format as follows: We are given $\mathbf{X}$, the features for all buildings in the CBECS dataset, and $\mathbf{e}$, the target `MFBTU` (energy) values for all buildings, where a row, $X_{i,:}$, represents the features for building i, and an entry, e_i, is the `MFBTU` value for building i. In the remainder of the paper we focus on predicting the logarithm of the actual `MFBTU` values as we have observed that the `MFBTU` values follow an approximate log-normal distribution, and some machine learning models will be able to better estimate the values in the log-transformed normal distribution [230, 231](see Figure A.3). Specifically, we let $y_i = \log_{10}(e_i)$, and refer to $\mathbf{y}$ as our target values. We want to learn a function, $f(X_{i,:}) = \hat{y}_i$, that takes the features of a building as input, and outputs the estimated log of the energy consumption for that building, $\hat{y}_i$. From this, we predict the `MFBTU` value for a building as $\hat{e}_i = 10^{\hat{y}_i}$. To estimate f we will use machine learning models such as: linear regression, gradient boosting regression models, and random forest regressors. In general, these models attempt to tune their internal parameters, θ, to minimize some loss function, L, between the target values and values predicted by the model, i.e. solving $\min_{\theta} L(\mathbf{y}, f(\mathbf{X}; \theta))$. The loss function will be a function that penalizes inaccurate measurements, for example, mean squared error, $MSE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$. Once a model is 'trained', and we are confident in its ability to *generalize* to unseen inputs, it can be used to estimate $\log_{10}(\texttt{MFBTU})$ values for buildings that are not in $\mathbf{X}$.

To evaluate the performance of the models on unseen data, i.e. to see how well the model is able to *generalize* to inputs it has not seen, we use stratified k-folds cross validation [232] on the CBECS dataset. This process involves splitting the data into k subsets, where each subset contains an equal portion of each class of building from the CBECS data, training the models on $k-1$ sets, then evaluating their performance on the single remaining testing set. This process is repeated k times so that each of the k sets is used as

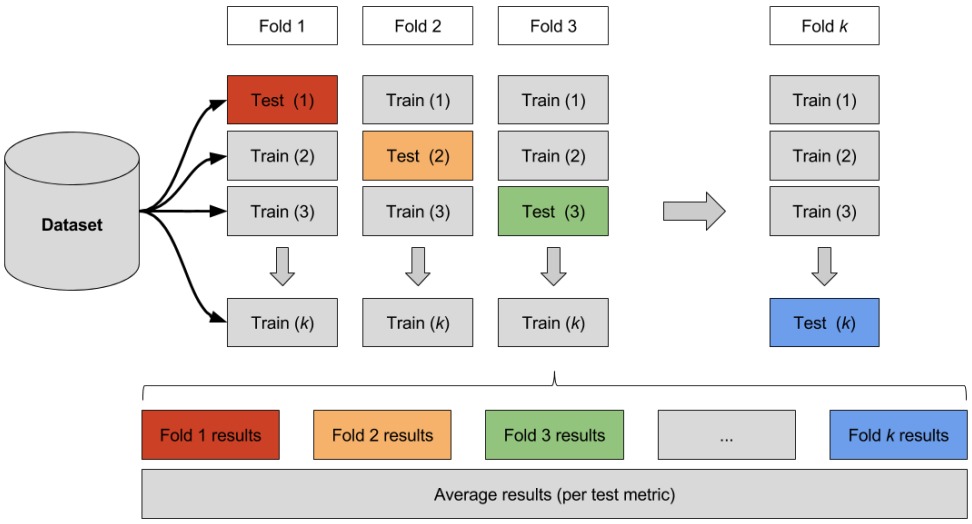


Figure A.5: Graphical representation of k-folds cross validation.

the testing set once. The evaluation metrics are reported as the average metric over the k iterations. Figure A.5 gives a graphical representation of this method. We chose stratified k-folds cross validation over the traditional k-folds cross validation because of the class imbalance in the CBECS dataset. The CBECS dataset contains 20 unique classes of buildings (where classes are defined as PBAs). The number of samples in each class differs from 1044 "Office Buildings" to 14 "Enclosed Malls", where each class has an unique energy consumption distribution, see Figures A.2 and A.3. This cross validation evaluation is the method used in the "Evaluation" step shown in Figure A.1. In all of our experiments we have set k equal to 10. During each cross validation split, we center and scale each feature in the training and testing splits based on statistics calculated from the training split (i.e. we subtract the mean and divide by the standard deviation). Finally, without loss of generality, we clip any negative predictions from any model to 0.

We use the following models in our experiments: linear regressor, ridge regressor, RBF kernel support vector regressor (SVR), elastic net regressor, linear kernel support vector regressor (linear SVR), adaboost regressor, bagging regressor, gradient boosting regressor (XGBoost), random forest regressor (RF regressor), extra trees regressor (ET regressor),

multi-layer perceptron regressor (MLP regressor), and k-nearest neighbor regressor (KNN regressor). All model implementations are based on the scikit-learn Python library [43] and use the default parameter settings. To evaluate the models we use stratified k-folds cross validation with $k = 10$, and record the cross-validated mean absolute error (mean AE), median absolute error (median AE), and the r^2 between the true and predicted $\log_{10}(\texttt{MFBTU})$ values. The r^2 values calculated between the predicted values, $\hat{\mathbf{y}}$, and the actual values, $\mathbf{y}$, is given as $r^2(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$, where $\bar{y}$ is the mean value of $\mathbf{y}$. A model which guesses the mean for every observation will have an r^2 score of 0, therefore any model which performs worse than this is "learning" the wrong relationships and will be useless. We also report the $10^{\text{Mean AE}}$ and $10^{\text{Median AE}}$, which capture the average number of multiples away the model's MFBTU estimate is from the true MFBTU value.

The CBECS dataset has many features that might not be feasible to collect in localized studies. The purpose of training models on the CBECS dataset is to later use them to estimate the energy consumption of commercial buildings in *any* city where there is building data, but no energy consumption data. Most features in the CBECS dataset such as, 'Number of Employees', 'Number of X-ray machines', or 'Insulation upgraded', are not commonly available, and therefore should not be included when training the models. It is important, however, to determine the influence of each possible feature included in the CBECS data on predicting energy consumption, in order to determine the potential benefits of additional data collection efforts. To this end, we run two sets of experiments using the methodology described in the previous two paragraphs: one that involves training the models using only a set of features that will be commonly available, or easily obtainable, in many cities and one that includes all of the features available in CBECS. We refer to the first group of features as the "common feature set"; it includes the following features: principal building activity, square feet, number of floors, heating degree days, and cooling degree days. We refer to the second group as the "extended feature set". As the "common feature set" is the set we expect to be available when using our models in specific urban

areas, Figure A.1 shows this set of features as common between the "Model Development" section and "Application" section.

To supplement the previous two experiments, and to aid the interpretability of our modeling process, we determine which features are the most important to the gradient boosting models (which we show are the best performing models). Feature importances in gradient boosting models are calculated as the amount of reduction in Gini impurity each feature causes over all splits for which that feature is present, over all of the trees that make up the model [157]. These values give us the relative importance of each feature included in a model, allowing us to rank the features in terms of "most useful" in the model, and compare the relative importance of features within a model. By performing this step for models trained with both the *common* and *extended* feature sets, we can see which features in the extended feature set that are not included in the common set will be most beneficial to include.

While machine learning models may perform well *within* the CBECS dataset, there is no guarantee from the cross-validated experimental results about the performance of the models on *external* data. Considering this, we validate our models on the augmented LL84 dataset, which describes the characteristics and energy consumption levels of 13,223 buildings in New York City. To do this we choose the best performing models from the first two experiments, train them with the CBECS data, then use them to predict the energy consumption values for each building in the LL84 dataset. Finally, in addition to this experiment, we perform a cross-validated experiment, with the same setup as our initial experiment on the CBECS data, using only the LL84 data (i.e. both training and testing models on the LL84 dataset). The results of this experiment will give us an upper bound on how well we can expect our models to perform on the LL84 data. The difference between these results, and the results of the previous experiment will show us how much information our general models are lacking about specific New York City energy consumption patterns.

A.4 Results and discussion

Our results are presented in five parts: 1.) testing the ability of machine learning models to predict commercial energy consumption from CBECS with the *common feature set* in Section A.4.1; 2.) testing the same models with the *extended feature set* from CBECS in Section A.4.2; 3.) evaluating the most important features from the previous two parts in Section A.4.3; 4.) validating our machine learning models on the LL84 dataset in Section A.4.4, and 5.) applying our machine learning models to the city of Atlanta in Section A.4.6. The methodology for all of these experiments can be found in Section A.3.

A.4.1 Experiments with common features

We first experiment with training machine learning models to predict commercial building energy consumption using a common set of features from the CBECS dataset. This *common feature set* contains only the features from CBECS that are also available in the augmented LL84 data, namely: principal building activity, square footage, number of floors, heating degree days, and cooling degree days. In addition to being common with the LL84 dataset (to allow for external validation), this set of features should be widely available for commercial buildings in many metropolitan areas through local or commercial datasets. This modeling choice will let the models we train with the CBECS data be applied to a wide range of metropolitan areas.

In Table A.3 we show the cross validated mean absolute error (mean AE), $10^{\text{mean AE}}$, median absolute error (median AE), $10^{\text{median AE}}$, and r^2 score of all models tested on the entire dataset. Because we perform a log_{10} transformation of the `MFBTU` values, $10^{\text{mean AE}}$ should be interpreted as the number of multiples away the predicted energy value would be from the true energy value for a building where our model exhibits the mean absolute error [3]. The same reasoning applies for the median absolute error. For all evaluation metrics,

[3] Given a mean or median absolute error, x, if for a building i, $x = |log_{10}(y_i) - log_{10}(\hat{y_i})| = |e_i - \hat{e_i}|$, then that means that $\frac{e}{\hat{e}} = 10^{\pm x}$.

we report the metric by averaging over the results of k folds, hence each metric also has a standard deviation σ associated with it over the folds, which we show as '+/- σ' in the results tables.

Table A.3 shows that over all classes of buildings, the gradient boosting regressor (XGBoost) outperforms the other models in all metrics, with an r^2 value of 0.82 and `MFBTU` predictions that are within 1.99 times of the true `MFBTU` value on average. The linear models (linear regression, ridge regression, etc.) all perform comparatively poorly, with a maximum r^2 score of 0.53.

In Figure A.6 we show a comparison of the predictions made by the linear regression model to those of the gradient boosting model. Qualitatively, this figure shows that the gradient boosting model is not systematically over- or under-estimating energy consumption, compared to the linear regression model. The linear regression models overestimate consumption at the low end of the actual energy consumption range, and underestimate at large energy consumption ranges. This unbiased attribute is important for models that will be used to create aggregate summaries of energy consumption. If we use a biased model to create building level predictions, that are then aggregated over some spatial areas (at the zipcode, or county level for example), any bias in the model's predictions will be compounded and will result in less accurate predictions.

Finally, in Table A.4 we show the resulting r^2 values for each model described in Section A.3.2 for each different class of building in the CBECS dataset. Consistent with the results observed in Table A.3, this table shows that the gradient boosting model is the best model over all classes of building, and generally performs better on classes with more training samples. For some classes of commercial buildings, such as "Service", and "Food Service", the per class predictions, using the *common feature set*, are particularly poor. This observation partially motivates our subsequent experiments in Sections A.4.2 and A.4.3, as we need to determine how the models should be improved to cover the deficiencies. Most of the models are unable to give better than average guesses on the smallest two classes,

"Refrigerated warehouse", and "Enclosed mall". This poor performance can possibly be explained by both the small number of samples for these classes, and the complexity of the 'features' that play a role in the determination of the true energy consumption of buildings in these categories. Buildings in the "Refrigerated warehouse" class, for example, will have expensive cooling equipment that make up the majority of their power consumption signal. Similarly, buildings in the "Laboratory" category are likely to have highly equipment with large energy demands, diminishing the impact of square footage as the most useful variable. For this reason the Federal Energy Management Program calculates the energy-savings goals of laboratories independent of square footage, which is the denominator of the goals of other building types [233].

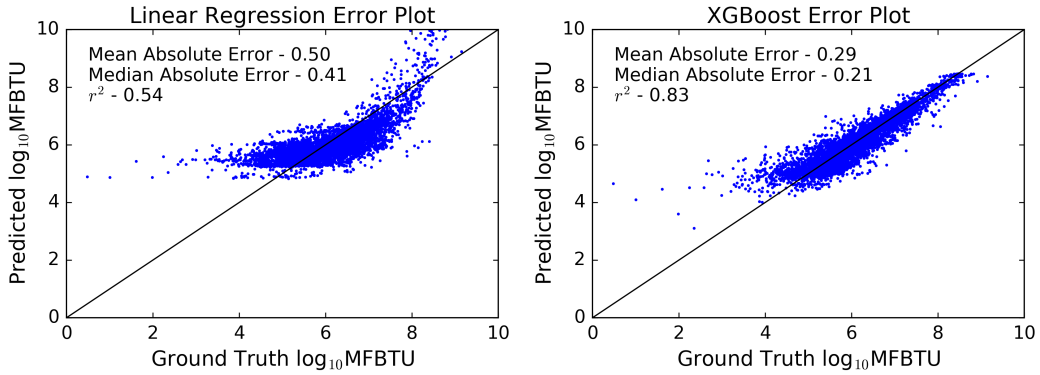


Figure A.6: Error plots comparing the predicted log of `MFBTU` values versus the true log values for the Linear Regression and XGBoost models. These models were trained on 9 out of 10 stratified splits, then used to predict log `MFBTU` values for all of the data points.

A.4.2 Experiments with extended features

This section involves repeating the experiments described in Section A.4.1 while using *all* of the features available in the CBECS dataset, i.e. the *extended feature set*.

In Table A.5 we show how the models perform using the extended feature set, comparable to the results in Table A.3. Here we see that the gradient boosting model again performs the best, with a 0.07 increase in r^2 over the same model trained using the common feature set. We also observe that the linear models perform much better, with nearly

equal results to those of the gradient boosting regressors. The large gap in performance between the linear regression models trained with the common feature set versus the ones trained with the extended features, compared to the relatively small gap between the gradient boosting models, further reinforces that the gradient boosting models should be used in external applications, where few features are available. This shows that gradient boosting models are able to combine the signals present in the common feature set to make predictions with accuracies that the linear regression models need many extra features to match. From this table we see that the mean and median absolute errors of the gradient boosting model also improve with the extended feature set. Considering the median absolute error, the gradient boosting model makes predictions within 1.48 multiples of the true `MFBTU` value. We discuss which features are most important to the gradient boosting models in Section A.4.3. In Table A.6 we show the r^2 of all the models per building type, similar to Table A.4. The gradient boosting model is still the best performing model in most cases. It is worse for buildings in the 'Food sales' and 'Public order and safety' classes, where the linear regression model is better. Linear regression models also tie for the best model in the 'Nonrefrigerated warehouse', 'Religious worship', 'Public assembly', and 'Retail other than mall' categories. With the common feature set, the gradient boosting model had an overall r^2 of 0.82, however performed very poorly in some classes of buildings, such as "Service", "Food Service", and "Nursing". With access to the *extended feature set*, the gradient boosting models were able to improve the r^2 scores for these classes of buildings by 0.29, 0.5, and 0.53 respectively. This supports our hypothesis that for some classes of buildings, more features than those found in the common feature set are needed to reliably predict a building's energy consumption. Examples might include behavioral variables reflecting how building equipment is utilized, which has become an increasing focus of energy-efficiency initiatives [234].

In general, these results give a rough upper bound on the ability of machine learning models to predict energy consumption from simple survey data. The gradient boosting

model is able to achieve a cross validated r^2 score of 0.89 when fit with the extended feature set of CBECS features, suggesting that it is able to generalize very well to unseen data. As we show in the next section, the models trained with the extended feature set available in CBECS can indicate which features should be prioritized in any data collection efforts, in order to close the performance gap between the models trained with the common feature set.

A.4.3 Feature importance

We have shown, without much surprise, that the gradient boosting models will perform better at predicting commercial building energy consumption when trained with the extended feature set, versus when trained with the common feature set. The question of 'which' features in the extended feature set are important motivates this experiment, as the answer to this question can guide future data collection efforts. We purposely keep our common feature set as simple as possible so that the trained models will be applicable to a wide range of metropolitan areas (with the assumption that many cities will be able to get access to these basic features). The most important features in the extended feature set that are not in the common set, should be the focus of data collection efforts, as they will give the largest boost to the models' predictive power.

In Table A.7 we show the top 10 most important features from the gradient boosting models trained with the CBECS data using the common feature set, and the extended feature set. From the most important features in the common feature set, we observe that the square footage of a commercial building is almost 3 times important as the next most important feature, and that the number of floors feature is relatively unimportant (possibly because it will be indirectly included in the square footage).

Out of the most important features out of the extended feature set, we observe that the square footage is still the most important feature, although the 'Number of employees' and 'Total hours open per week' are the second and third most important features, both

of which are not present in the common feature set. This idea, that building occupancy is an important feature, is supported up by a recent study that shows that occupancy rates of commercial buildings are important to consider in energy savings measures [235]. The climate related features ("heating degree days", "cooling degree days") are relatively less important. Similarly, other features that have shown to be important to energy consumption calculations, such as building envelope insulating characteristics [236](e.g. wall and roof constriction materials), are not included in the model's top ten important features. We note that the important variables are determined based on how much they contribute to the model's decision, and are not necessarily directly related to the actual calculation of building energy consumption. This relationship is illustrated by looking at the most important features for estimated energy consumption within buildings of a given principal building activity. From Table A.6 we observed that in the 'Service', 'Religious worship' 'Food Service', 'Vacant', 'Other', and 'Nursing' classes of buildings, adding all the features increased the r^2 score of the gradient boosting model by over 0.2. When we train a gradient boosting regressor on *just* samples from the 'Service' class of buildings we observe that the most important feature is 'Total hours open per week', instead of 'Square footage'. This suggests that for some PBAs, the common feature set does not contain the correct signals to reproduce the `MFBTU` targets. This also suggests that within class consumption differences are explained by features that are not necessarily relevant to all classes. For further results on the features that are most important per PBA, see the"Feature importances" notebook in the accompanying GitHub repository.

A.4.4 Validation

The CBECS data provides us with a national dataset of commercial buildings that we can train and test our models with, however this does not let us make conclusions about the efficacy of the models for predicting energy consumption in a particular metropolitan area. We validate our models by using them to predict the energy consumption values of buildings

in New York City, for which the true consumption values are known.

We train a gradient boosting model on *all* of the CBECS data with the common feature set, then apply that model on the 2,612 commercial buildings from the augmented LL84 dataset and record the same metrics from previous experiments. This results in a mean absolute error of 0.25, median absolute error of 0.15, and r^2 value of 0.50. Consistent with the interpretation given in Section A.4.1, the predicted energy consumption values have a mean of 1.78 multiples away from the true value, and a median of 1.41 multiples away from the true value. These mean and median errors are *better* than the best values observed from the CBECS dataset (both in the reduced features, and all features cases), although the r^2 fit is worse.

We further train and test all the machine learning models on solely the LL84 dataset using the same methodology as in the first two experiments. The results from this test are shown in Table A.8. The gradient boosting model is again the best performing model, however, surprisingly, when the gradient boosting model is trained on the LL84 data, it only performs slightly better than the model that was trained on the CBECS data. Specifically, the gradient boosting model has an r^2 of 0.54 from training on the LL84 data, compared to an r^2 value of 0.51 from training on the CBECS data. The mean absolute error is 0.01 lower, and the median absolute error is the same. These results provide strong evidence that our model trained with the CBECS data is able to be applied to specific metropolitan areas while maintaining reasonable results.

A.4.5 Extended validation

Here we extend our discussion from the previous section on the validity of our models when applied to specific metropolitan areas. Specifically we exam the errors made by XGBoost model, that was trained on all of the CBECS data, and applied to the Augmented Local Law 84 Dataset. Figure A.7 shows a scatter plot of the predicted $\log_{10}(\texttt{MFBTU})$ values versus the actual values for all points in the LL84 dataset. From this plot we can see that

although the majority of the predicted values are similar to their actual values, there are a handful of points that are badly over- and under-estimated. A common feature among these bad outlier predictions, is that they all have extreme Energy Star scores. Figure A.8 shows the model's error versus the reported Energy Star score (from the LL84 data), and illustrates that the "mistakes" being made by the XGBoost model can be explained through a building's Energy Star score. An Energy Star score is a value between 1 and 100 that is calculated by the EPA based on the most recent CBECS data, and represents how energy efficient a building is compared to the national average. A score of 50 means that a building is as energy efficient as 50% of other buildings within its category (based on principal building activity) nationally. Similarly, a score of 99 means that a building is more efficient than 99% of other buildings within in its category. Our model, which is trained with the common feature set, makes large errors for buildings that fall on either end of this extreme, over predicting the energy consumption of buildings that are very energy efficient, and under predicting the consumption of buildings that have poor energy efficiency. A building's Energy Star score is calculated using features from the extended feature set, and considering that all of the models we test perform better when using more features, we believe that these features are able to explain the errors observed in the LL84 data. We are unable to test this hypothesis however, as the LL84 data does not include the richer set of features used to calculate the Energy Star scores.

A.4.6 Case study in Atlanta

We show how our models can be applied to a large metropolitan area by creating commercial energy consumption summaries for the 20 county Atlanta metropolitan area. To do this, we applied the CBECS-trained gradient boosting regression model to the 73,388 commercial buildings in Atlanta from the CoStar real estate database[4]. We supplement the CoStar data with the 2017 heating and cooling degree day data from the Oak Ridge Climate models

[4]This dataset is continuously updated, see `http://www.costar.com/`. The dataset that we use was downloaded in March of 2017.

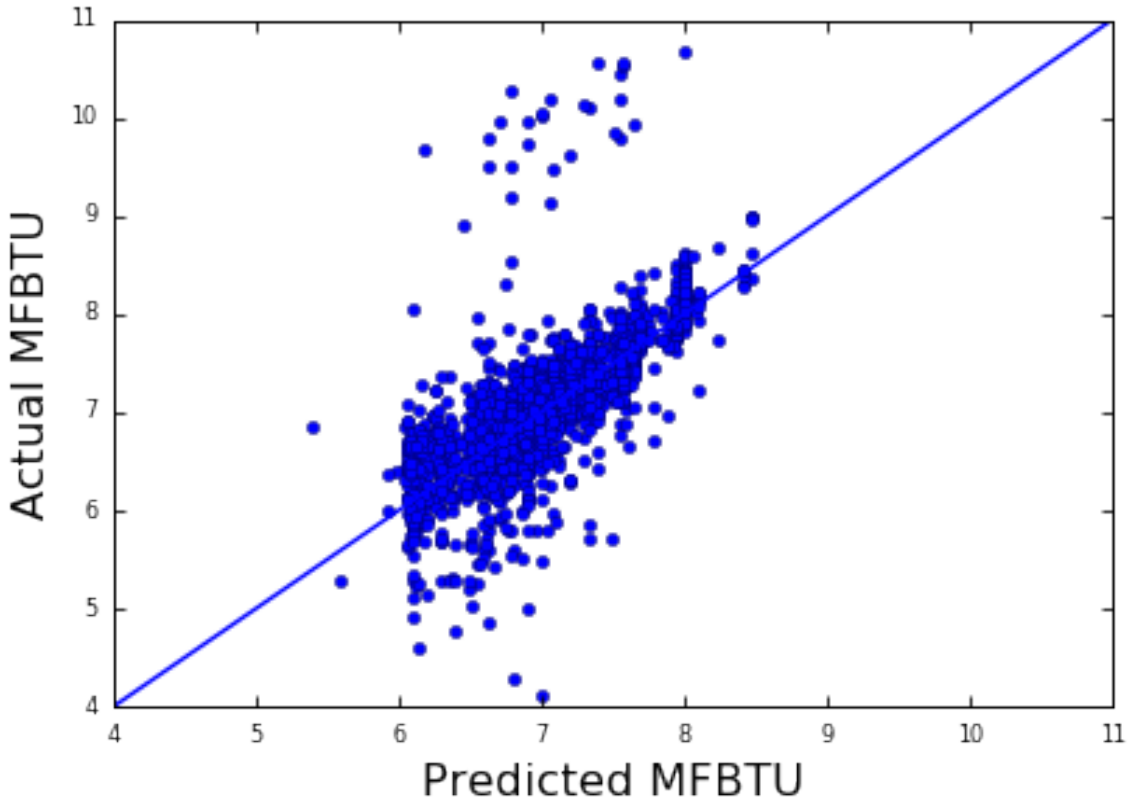


Figure A.7: Error plots comparing the predicted log of `MFBTU` values versus the true log values for the XGBoost model on the LL84 validation dataset.

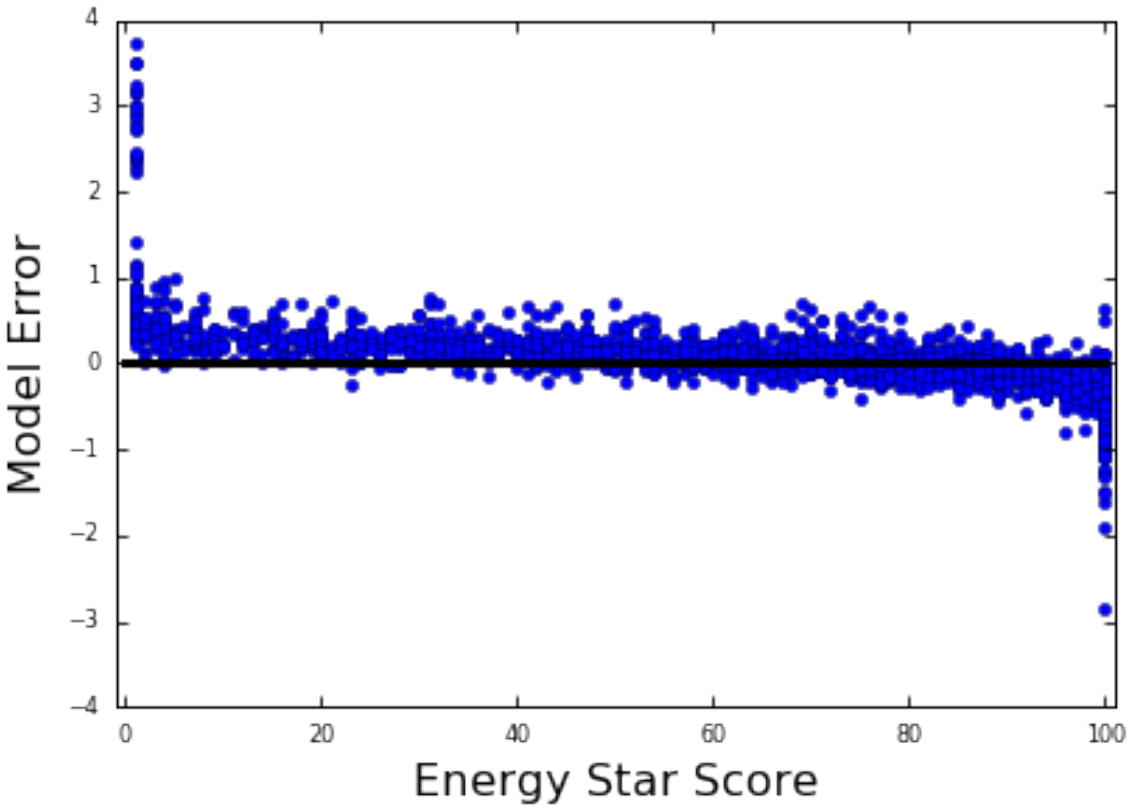


Figure A.8: XGBoost model error versus EnergyStar score.

(using the same methodology we used to create the Augmented LL84 dataset). Figure A.9 shows a map of the *total estimated energy consumption* values from all commercial buildings aggregated at the Transportation Analysis Zone (TAZ) level, and a map of the *median estimated energy use intensity across commercial buildings* in each TAZ[5]. Both maps are colored according to quantile binning with 10 bins. As we show in Figure A.6, the gradient boosting model does not systematically overestimate or underestimate energy consumption values. Considering this, we expect the modeling errors for individual building's energy consumptions to cancel out in the aggregate energy consumption estimates. From the energy consumption map we observe that the greatest energy consuming TAZs are clustered in the "Downtown" and "Midtown" parts of Atlanta (in the center of the mapped area), as well as the suburban cities surrounding Atlanta. Commercial energy consumption is generally greater in TAZs immediately adjacent to the I-85 and I-75 highways that cut diagonally through the city, from southwest to northeast, and southeast to northwest, respectively. The median energy use intensity map shows that, although TAZs in the "Downtown" and "Midtown" parts of Atlanta consume more energy, they are more energy efficient on average than TAZs in the northern Buckhead suburb.Similarly, these maps show some TAZs in the surrounding suburban cities that have disproportionately high energy consumption compared to their surroundings, indicating locations where energy efficiency building retrofits should be considered [237]. Our total estimated commercial energy consumption for Atlanta is 126.62 billion kBTUs/year, which would make up 0.7% of the total annual commercial energy consumption of the U.S. in 2016 [238].

We note that the city of Atlanta's new energy benchmarking ordinance for commercial buildings may change this geography of energy consumption in commercial buildings. It aims to achieve a 20% reduction of energy consumption in Atlanta's private and City-owned buildings over 25,000 square feet, by 2030 [239]. If successful, this could curb the energy consumption peaks shown in the Downtown and Midtown TAZs.

[5]The energy use intensity of a building is its total energy consumption divided by its square footage.

Predicting the commercial building energy consumption landscape in Atlanta is just a single example of what our models are capable of doing. The results illustrate the ability to use our US-wide CBECS-based commercial building energy consumption models for various kinds of analysis and scenario evaluations that can inform urban planning and policy making. Our models can be applied to other metropolitan areas for which there is building level data available by following the instructions given in our GitHub repository[6].

A.5 Conclusion

We create machine learning models trained on the CBECS dataset for estimating commercial building energy consumption, analyze feature importance in our models, then validate the models on external data from New York City, and create commercial building energy consumption estimates for the Atlanta metropolitan area. An important aspect of our work involves limiting the information about each building used by the machine learning models to five *commonly available* features, so that our models can be used in a wide range of metropolitan areas without requiring expensive data collection efforts. We find that some of the models are able to perform acceptably well under this constraint, and the gradient boosting models are able to make predictions that are on average under 1.78 multiples away from the true value on the external validation dataset. Although this error is too large for analyzing the energy consumption of any specific building, when the models are used to make predictions for all the buildings in entire metropolitan areas (where individual prediction errors will cancel out when aggregated), as we show for Atlanta, they can offer useful insights into a city's commercial energy consumption landscape. Furthermore, our analysis of important features used by the machine learning models will serve to drive future data collection efforts that could help maximize the accuracy of the models.

Admittedly, our modeling approach has several limitations. One limitation is that our models can only be used in the United States, as the CBECS training dataset only provides

[6] `https://github.com/SEI-ENERGY/Commercial-Energy`

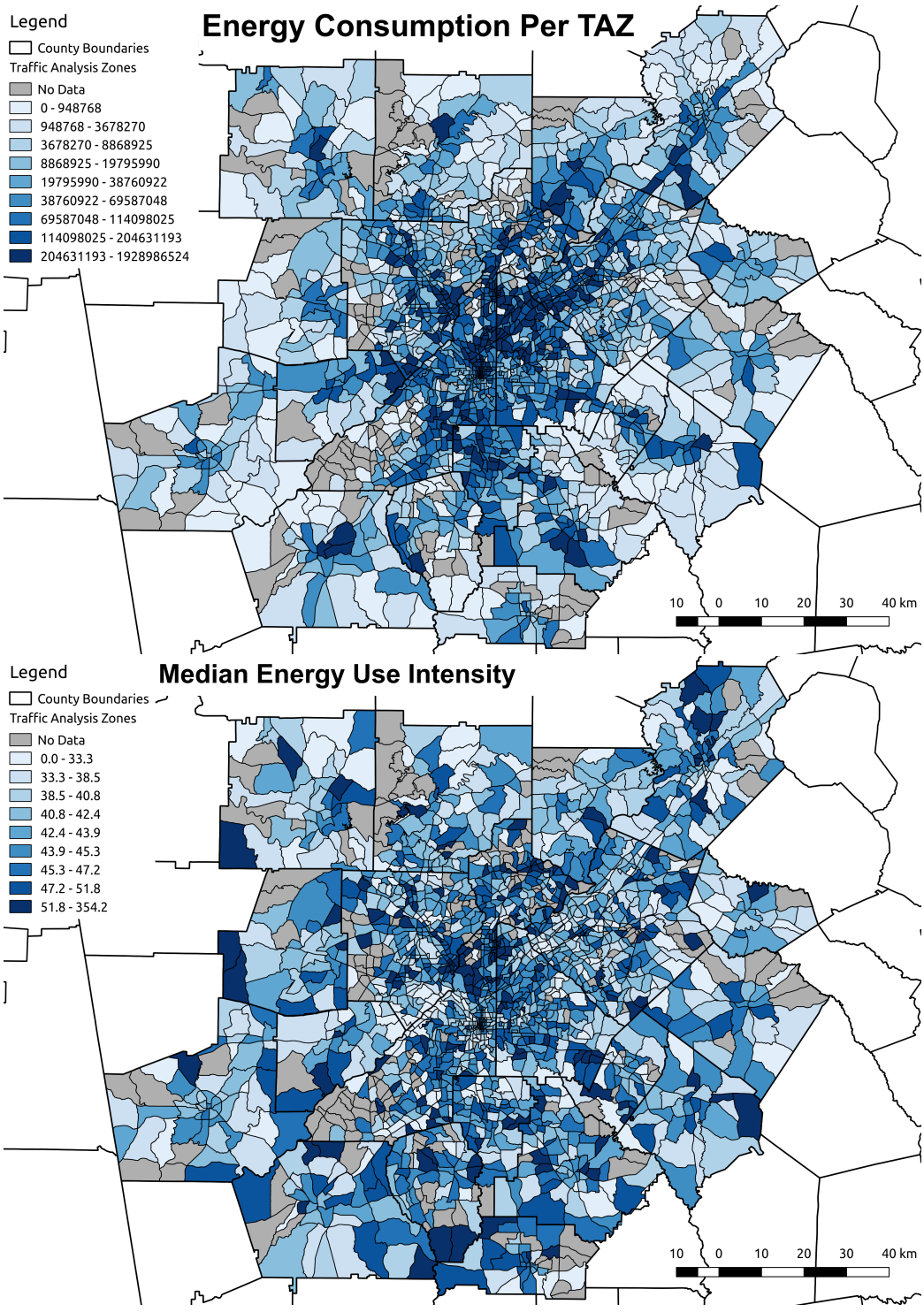


Figure A.9: Estimated energy consumption (kBTU/year) of commercial buildings in Atlanta, aggregated per TAZ (**top**). Median estimated energy use intensity of commercial buildings per TAZ (**bottom**).

a validated statistical representation of commercial buildings in the US. When more detailed commercial building datasets become available in other countries, the same methodology we use in this paper will be applicable. Another limitation of our models is the trade-off between accuracy and data requirements which we show with the results comparing the accuracy between models using our *common feature set* and *extended feature set*. Using more building level variables when modeling commercial building energy consumption predictably gives better results, however will limit the usefulness of the models in metropolitan areas that do not have access to such detailed data. While we validate the results of our models trained with "common features" using data from New York City, and show they give reasonable results, the models will not be able to capture the energy consumption patterns of buildings that are extremely efficient or inefficient.

Related works in statistical building energy consumption predictions have not been thoroughly explored using more complicated machine learning models, such as gradient boosting regression models (the best performing model). Our results show that these higher capacity models are more capable of exploiting a limited number of features to achieve better performance than possible with general linear models. Currently, city planners and policy makers will be able to use our models to create summary commercial energy consumption maps to assist with future planning, and achieving sustainable development goals. As more energy consumption data becomes available through crucial data collection efforts such as New York City's Local Law 84, statistical models for estimating building energy consumption will become more powerful, and will be able to give more confident estimates, further improving the capability for understanding our urban environments.

Our work has several opportunities for future development that we are interested in pursuing. One important aspect of machine learning modeling is model selection. Many of the machine learning models that we reference in this paper have hyperparameters that can be tuned to further increase their predictive ability. As maximizing the predictive performance of any specific model was not the focus of this paper, we opted to leave the hyperparameter

settings at their default values, and instead focus on the role of available features in model performance. We are interested in performing a broad computational study that focuses on maximizing model performance on the task of predicting building energy consumption. Secondly, we are interested in applying the models developed in this study to all the commercial buildings in major metropolitan areas as part of a summary of total metropolitan energy consumption. Finally, an analysis of the potential impact of future climate scenarios and alternative patterns of urban growth can be informed by the models developed in this paper. We are interested in using climate model projections to evaluate how the energy consumption landscapes of different cities will change under various climate scenarios.

Table A.2: Mapping between the LL84 'Primary Building Activity' and the CBECS 'Principal Building Activity' fields. We exclude the "Not Available", "Multifamily Housing", "Manufacturing/Industrial Plant", "Parking", and "Mixed Use Property" classes from the LL84 data.

LL84 PBA	Matched CBECS PBA	CBECS PBA Code
Courthouse	Office	2
Bank Branch	Office	2
Financial Office	Office	2
Medical Office	Office	2
Office	Office	2
Laboratory	Laboratory	4
Self-Storage Facility	Nonrefrigerated warehouse	5
Distribution Center	Nonrefrigerated warehouse	5
Non-Refrigerated Warehouse	Nonrefrigerated warehouse	5
Wholesale Club/Supercenter	Food sales	6
Restaurant	Food sales	6
Supermarket/Grocery Store	Food sales	6
Police Station	Public order and safety	7
Other - Public Services	Public order and safety	7
Urgent Care/Clinic/Other Outpatient	Outpatient health care	8
Outpatient Rehabilitation/Physical Therapy	Outpatient health care	8
Refrigerated Warehouse	Refrigerated warehouse	11
Data Center	Refrigerated warehouse	11
Worship Facility	Religious worship	12
Museum	Public assembly	13
Other - Entertainment/Public Assembly	Public assembly	13
Social/Meeting Hall	Public assembly	13
Fitness Center/Health Club/Gym	Public assembly	13
Senior Care Community	Public assembly	13
Library	Public assembly	13
Movie Theater	Public assembly	13
Lifestyle Center	Public assembly	13
College/University	Education	14
Adult Education	Education	14
Other - Education	Education	14
K-12 School	Education	14
Hospital (General Medical & Surgical)	Inpatient health care	16
Other - Specialty Hospital	Inpatient health care	16
Ambulatory Surgical Center	Inpatient health care	16
Residential Care Facility	Inpatient health care	16
Hotel	Lodging	18
Other - Lodging/Residential	Lodging	18
Residence Hall/Dormitory	Lodging	18
Strip Mall	Strip shopping mall	23
Other - Mall	Enclosed mall	24
Enclosed Mall	Enclosed mall	24
Automobile Dealership	Retail other than mall	25
Retail Store	Retail other than mall	25
Other - Services	Service	26
Repair Services (Vehicle, Shoe, Locksmith, etc.)	Service	26
Personal Services (Health/Beauty, Dry Cleaning, etc.)	Service	26
Performing Arts	Other	91
Other	Other	91
Other - Recreation	Other	91
Other - Utility	Other	91

Table A.3: **Common features**. Results of all machine learning models trained on the common feature set. The mean absolute error (mean AE), median absolute error (median AE), and the r^2 values are calculated in terms of log_{10} MFBTU values. The $10^{\text{Mean AE}}$ and $10^{\text{Median AE}}$ columns show the average number of multiples away the model's estimate is from the true value. The values following "+/-" are the standard deviations of each metric calculated over the 10 cross validation folds.

Common Features	**Mean Absolute Error**	$10^{\textbf{Mean AE}}$	**Median Absolute Error**	$10^{\textbf{Median AE}}$	r^2
XGBoost	**0.30 +/- 0.01**	**1.99 +/- 0.06**	**0.22 +/- 0.01**	**1.66 +/- 0.03**	**0.82 +/- 0.02**
Bagging	0.33 +/- 0.01	2.13 +/- 0.07	0.24 +/- 0.01	1.73 +/- 0.05	0.78 +/- 0.03
MLP Regressor	0.33 +/- 0.01	2.16 +/- 0.05	0.25 +/- 0.01	1.77 +/- 0.04	0.78 +/- 0.02
Random Forest Regressor	0.33 +/- 0.02	2.13 +/- 0.07	0.24 +/- 0.01	1.73 +/- 0.05	0.78 +/- 0.02
Extra Trees Regressor	0.34 +/- 0.02	2.17 +/- 0.08	0.24 +/- 0.01	1.74 +/- 0.05	0.76 +/- 0.03
SVR	0.39 +/- 0.01	2.44 +/- 0.07	0.29 +/- 0.01	1.95 +/- 0.04	0.70 +/- 0.03
KNN Regressor	0.43 +/- 0.01	2.68 +/- 0.08	0.32 +/- 0.02	2.10 +/- 0.07	0.65 +/- 0.03
AdaBoost	0.43 +/- 0.03	2.71 +/- 0.16	0.36 +/- 0.03	2.29 +/- 0.17	0.68 +/- 0.03
Linear SVR	0.51 +/- 0.02	3.28 +/- 0.15	0.40 +/- 0.02	2.54 +/- 0.11	0.52 +/- 0.04
Linear Regression	0.52 +/- 0.02	3.33 +/- 0.13	0.43 +/- 0.02	2.72 +/- 0.12	0.53 +/- 0.03
Ridge Regressor	0.52 +/- 0.02	3.33 +/- 0.13	0.43 +/- 0.02	2.72 +/- 0.12	0.53 +/- 0.03
ElasticNet	0.76 +/- 0.02	5.75 +/- 0.32	0.67 +/- 0.03	4.67 +/- 0.35	0.09 +/- 0.01
Lasso	0.79 +/- 0.02	6.17 +/- 0.35	0.69 +/- 0.03	4.92 +/- 0.38	0.00 +/- 0.00

Table A.4: **Common features, per PBA**. Prediction accuracy is broken down by PBA. This table shows the r^2 scores of the predicted values by the top 5 performing models and the Linear Regression model trained with the *common feature set*. The values following "+/-" are the standard deviations of each metric calculated over the 10 cross validation folds.

	n	Linear Regression	ET Regressor	RF Regressor	Bagging	MLP Regressor	XGBoost
Office	1044	0.45 +/- 0.05	0.85 +/- 0.03	0.86 +/- 0.03	0.86 +/- 0.03	0.84 +/- 0.03	**0.88 +/- 0.02**
Education	580	0.37 +/- 0.05	0.80 +/- 0.04	0.80 +/- 0.04	0.81 +/- 0.04	0.80 +/- 0.04	**0.84 +/- 0.03**
Nonrefrigerated warehouse	567	0.31 +/- 0.07	0.55 +/- 0.09	0.55 +/- 0.11	0.55 +/- 0.12	0.59 +/- 0.05	**0.63 +/- 0.06**
Service	354	0.08 +/- 0.10	0.12 +/- 0.25	0.22 +/- 0.19	0.25 +/- 0.20	0.31 +/- 0.12	**0.37 +/- 0.16**
Religious worship	322	0.12 +/- 0.09	0.39 +/- 0.29	0.45 +/- 0.21	0.43 +/- 0.27	0.46 +/- 0.09	**0.57 +/- 0.16**
Retail other than mall	316	0.17 +/- 0.11	0.69 +/- 0.12	0.70 +/- 0.11	0.70 +/- 0.11	0.68 +/- 0.11	**0.73 +/- 0.11**
Public assembly	311	0.42 +/- 0.10	0.73 +/- 0.06	0.77 +/- 0.04	0.77 +/- 0.04	0.75 +/- 0.06	**0.81 +/- 0.03**
Food service	306	negative	negative	negative	negative	0.07 +/- 0.08	**0.20 +/- 0.12**
Strip shopping mall	277	0.32 +/- 0.07	0.67 +/- 0.11	0.67 +/- 0.12	0.67 +/- 0.12	0.70 +/- 0.08	**0.73 +/- 0.09**
Lodging	221	0.40 +/- 0.15	0.81 +/- 0.12	0.79 +/- 0.12	0.79 +/- 0.11	0.77 +/- 0.15	**0.83 +/- 0.11**
Inpatient health care	215	negative	**0.81 +/- 0.06**	0.80 +/- 0.07	0.79 +/- 0.07	0.80 +/- 0.07	**0.81 +/- 0.07**
Outpatient health care	131	0.21 +/- 0.24	0.69 +/- 0.24	0.72 +/- 0.20	0.72 +/- 0.20	0.64 +/- 0.23	**0.76 +/- 0.16**
Food sales	111	0.03 +/- 0.17	0.46 +/- 0.24	0.45 +/- 0.21	0.48 +/- 0.19	0.45 +/- 0.22	**0.57 +/- 0.23**
Vacant	101	negative	negative	negative	negative	0.06 +/- 0.48	**0.17 +/- 0.45**
Other	68	negative	negative	negative	negative	negative	**0.23 +/- 0.79**
Nursing	62	**0.31 +/- 0.27**	**0.31 +/- 0.79**	0.20 +/- 1.14	0.21 +/- 1.05	0.26 +/- 1.01	0.25 +/- 1.28
Public order and safety	60	0.24 +/- 0.37	0.58 +/- 0.50	0.58 +/- 0.57	0.56 +/- 0.56	0.65 +/- 0.20	**0.69 +/- 0.34**
Laboratory	23	negative	0.54 +/- 0.43	0.33 +/- 0.55	0.26 +/- 0.64	0.11 +/- 0.98	**0.59 +/- 0.52**
Refrigerated warehouse	16	negative	negative	negative	negative	negative	negative
Enclosed mall	14	negative	negative	negative	0.05 +/- 0.99	negative	**0.17 +/- 0.31**
Total	5099	0.53 +/- 0.03	0.76 +/- 0.03	0.78 +/- 0.02	0.78 +/- 0.03	0.78 +/- 0.02	**0.82 +/- 0.02**

Table A.5: **Extended features**. Results of all machine learning models trained/tested with the extended feature set, compared to the XGBoost model results from Table A.3. The mean absolute error (mean AE), median absolute error (median AE), and the r^2 values are calculated in terms of log_{10} MFBTU values. The $10^{\text{Mean AE}}$ and $10^{\text{Median AE}}$ columns show the average number of multiples away the model's estimate is from the true value.

Extended Features	**Mean Absolute Error**	$10^{\textbf{Mean AE}}$	**Median Absolute Error**	$10^{\textbf{Median AE}}$	r^2
XGBoost with Common Features	0.30 +/- 0.01	1.99 +/- 0.06	0.22 +/- 0.01	1.66 +/- 0.03	0.82 +/- 0.02
XGBoost	**0.23 +/- 0.01**	**1.69 +/- 0.02**	**0.17 +/- 0.01**	**1.48 +/- 0.03**	**0.89 +/- 0.01**
Linear Regression	0.24 +/- 0.01	1.75 +/- 0.02	0.19 +/- 0.01	1.53 +/- 0.04	0.88 +/- 0.01
Ridge Regressor	0.24 +/- 0.01	1.75 +/- 0.02	0.19 +/- 0.01	1.53 +/- 0.04	0.88 +/- 0.01
SVR	0.25 +/- 0.01	1.79 +/- 0.04	0.19 +/- 0.01	1.53 +/- 0.03	0.87 +/- 0.01
Bagging	0.25 +/- 0.01	1.79 +/- 0.04	0.18 +/- 0.01	1.53 +/- 0.04	0.87 +/- 0.02
Random Forest Regressor	0.25 +/- 0.01	1.79 +/- 0.04	0.18 +/- 0.01	1.53 +/- 0.04	0.87 +/- 0.01
Extra Trees Regressor	0.25 +/- 0.01	1.79 +/- 0.04	0.19 +/- 0.01	1.54 +/- 0.03	0.87 +/- 0.01
Linear SVR	0.26 +/- 0.01	1.80 +/- 0.03	0.20 +/- 0.01	1.58 +/- 0.04	0.87 +/- 0.01
AdaBoost	0.32 +/- 0.01	2.07 +/- 0.05	0.26 +/- 0.01	1.80 +/- 0.05	0.82 +/- 0.01
KNN Regressor	0.37 +/- 0.01	2.34 +/- 0.06	0.29 +/- 0.01	1.93 +/- 0.04	0.75 +/- 0.02
MLP Regressor	0.45 +/- 0.02	2.82 +/- 0.11	0.36 +/- 0.02	2.31 +/- 0.10	0.64 +/- 0.03
ElasticNet	0.60 +/- 0.02	4.00 +/- 0.20	0.51 +/- 0.02	3.26 +/- 0.16	0.40 +/- 0.01
Lasso	0.79 +/- 0.02	6.17 +/- 0.35	0.69 +/- 0.03	4.92 +/- 0.38	0.00 +/- 0.00

Table A.6: **Extended features, per PBA**. Prediction accuracy is broken down by PBA. This table shows the r^2 scores of the predicted values by the top 4 performing models and the Linear Regression model, trained/tested with the extended feature set, compared to the XGBoost model results from Table A.4.

	n	Linear Regression	RF Regressor	Bagging	MLP Regressor	XGBoost	XGBoost with Common Features
Office	1044	0.90 +/- 0.01	0.90 +/- 0.02	0.90 +/- 0.02	0.66 +/- 0.07	**0.91 +/- 0.01**	0.88 +/- 0.02
Education	580	0.84 +/- 0.03	0.85 +/- 0.03	0.85 +/- 0.03	0.34 +/- 0.16	**0.87 +/- 0.02**	0.84 +/- 0.03
Nonrefrigerated warehouse	567	**0.81 +/- 0.03**	0.77 +/- 0.06	0.77 +/- 0.06	0.59 +/- 0.10	**0.81 +/- 0.04**	0.63 +/- 0.06
Service	354	0.65 +/- 0.11	0.61 +/- 0.12	0.60 +/- 0.12	0.14 +/- 0.27	**0.66 +/- 0.10**	0.37 +/- 0.16
Religious worship	322	**0.77 +/- 0.07**	0.72 +/- 0.08	0.72 +/- 0.09	0.34 +/- 0.35	**0.77 +/- 0.09**	0.57 +/- 0.16
Retail other than mall	316	**0.86 +/- 0.05**	0.81 +/- 0.08	0.81 +/- 0.08	0.45 +/- 0.19	**0.86 +/- 0.06**	0.73 +/- 0.11
Public assembly	311	**0.89 +/- 0.03**	0.86 +/- 0.04	0.86 +/- 0.03	0.60 +/- 0.15	**0.89 +/- 0.02**	0.81 +/- 0.03
Food service	306	0.66 +/- 0.07	0.56 +/- 0.13	0.56 +/- 0.15	negative	**0.70 +/- 0.09**	0.20 +/- 0.12
Strip shopping mall	277	0.85 +/- 0.06	0.87 +/- 0.05	0.87 +/- 0.04	0.21 +/- 0.27	**0.91 +/- 0.03**	0.73 +/- 0.09
Lodging	221	0.85 +/- 0.07	0.82 +/- 0.11	0.83 +/- 0.11	0.24 +/- 0.27	**0.87 +/- 0.07**	0.83 +/- 0.11
Inpatient health care	215	0.71 +/- 0.12	**0.84 +/- 0.06**	**0.84 +/- 0.07**	negative	**0.84 +/- 0.08**	0.81 +/- 0.07
Outpatient health care	131	0.83 +/- 0.10	0.82 +/- 0.11	0.82 +/- 0.11	0.45 +/- 0.29	**0.84 +/- 0.11**	0.76 +/- 0.16
Food sales	111	**0.74 +/- 0.17**	0.61 +/- 0.24	0.60 +/- 0.26	negative	0.68 +/- 0.19	0.57 +/- 0.23
Vacant	101	0.40 +/- 0.19	0.29 +/- 0.39	0.23 +/- 0.53	negative	**0.48 +/- 0.20**	0.17 +/- 0.45
Other	68	0.61 +/- 0.30	0.52 +/- 0.53	0.54 +/- 0.49	negative	**0.64 +/- 0.28**	0.23 +/- 0.79
Nursing	62	0.71 +/- 0.21	0.56 +/- 0.72	0.55 +/- 0.77	negative	**0.78 +/- 0.22**	0.25 +/- 1.28
Public order and safety	60	**0.83 +/- 0.12**	0.77 +/- 0.24	0.78 +/- 0.23	negative	0.80 +/- 0.21	0.69 +/- 0.34
Laboratory	23	0.58 +/- 0.32	0.16 +/- 0.97	0.17 +/- 0.94	negative	0.45 +/- 0.70	**0.59 +/- 0.52**
Refrigerated warehouse	16	negative	negative	negative	negative	negative	negative
Enclosed mall	14	negative	negative	negative	negative	negative	**0.17 +/- 0.31**
Total	5099	0.88 +/- 0.01	0.87 +/- 0.01	0.87 +/- 0.02	0.64 +/- 0.03	**0.89 +/- 0.01**	0.82 +/- 0.02

Table A.7: Top 10 important features for the XGBoost model trained with all data using both the common and extended feature sets.

Common Feature Set			Extended Feature Set		
Feature Name	**Feature Description**	**Importance**	**Feature Name**	**Feature Description**	**Importance**
SQFT	Square footage	0.3634	SQFT	Square footage	0.1391
CDD65	Cooling degree days (base 65)	0.1153	NWKER	Number of employees	0.0576
HDD65	Heating degree days (base 65)	0.1125	WKHRS	Total hours open per week	0.0557
PBA 5	Non-refrigerated warehouse	0.0569	ZMFBTU	Imputed major fuels consumption	0.0312
PBA 1	Vacant	0.0524	MONUSE	Months in use	0.0299
PBA 6	Food sales	0.0412	NGUSED	Natural gas used	0.0295
PBA 15	Food service	0.0384	HDD65	Heating degree days (base 65)	0.0293
PBA 23	Strip shopping mall	0.0348	HEATP	Percent heated	0.0278
PBA 12	Religious worship	0.0345	CDD65	Cooling degree days (base 65)	0.0224
PBA 4	Laboratory	0.0282	NWKERC	Number of employees category	0.0221

Table A.8: **LL84 Validation**. Comparison of the best external model tested on the LL84 dataset (out of sample validation result) to all machine learning models trained and tested on the LL84 dataset. The first row, 'XGBoost - CBECS', is the best external model and shows the results from applying the XGBoost model trained on all of the CBECS data, to all of the LL84 data. The remaining rows show the cross validated results on models trained and tested on the LL84 dataset. All results are shown with models using the common feature set. The mean absolute error (mean AE), median absolute error (median AE), and the r^2 values are calculated in terms of log_{10} `MFBTU` values. The $10^{\text{Mean AE}}$ and $10^{\text{Median AE}}$ columns show the average number of multiples away the model's estimate is from the true value.

	Mean Absolute Error	$10^{\textbf{Mean AE}}$	**Median Absolute Error**	$10^{\textbf{Median AE}}$	r^2
XGBoost - CBECS	0.25	1.78	0.15	1.41	0.51
XGBoost	**0.24 +/- 0.02**	**1.75 +/- 0.09**	**0.15 +/- 0.01**	**1.40 +/- 0.03**	**0.54 +/- 0.09**
SVR	0.25 +/- 0.02	1.77 +/- 0.10	**0.15 +/- 0.01**	**1.40 +/- 0.03**	0.51 +/- 0.11
Linear SVR	0.28 +/- 0.02	1.92 +/- 0.08	0.17 +/- 0.00	1.50 +/- 0.01	0.42 +/- 0.05
MLP Regressor	0.28 +/- 0.04	1.92 +/- 0.17	0.17 +/- 0.02	1.48 +/- 0.06	0.44 +/- 0.13
Linear Regression	0.29 +/- 0.02	1.96 +/- 0.10	0.19 +/- 0.01	1.56 +/- 0.05	0.44 +/- 0.08
Ridge Regressor	0.29 +/- 0.02	1.96 +/- 0.10	0.19 +/- 0.01	1.56 +/- 0.05	0.44 +/- 0.08
Bagging	0.29 +/- 0.02	1.95 +/- 0.09	0.18 +/- 0.01	1.50 +/- 0.04	0.43 +/- 0.08
Random Forest Regressor	0.29 +/- 0.02	1.95 +/- 0.10	0.18 +/- 0.02	1.51 +/- 0.05	0.43 +/- 0.08
Extra Trees Regressor	0.30 +/- 0.03	2.00 +/- 0.12	0.18 +/- 0.01	1.51 +/- 0.05	0.39 +/- 0.09
KNN Regressor	0.30 +/- 0.03	2.01 +/- 0.15	0.19 +/- 0.02	1.53 +/- 0.06	0.40 +/- 0.12
AdaBoost	0.42 +/- 0.07	2.67 +/- 0.43	0.30 +/- 0.04	2.01 +/- 0.20	0.14 +/- 0.22
Lasso	0.45 +/- 0.01	2.80 +/- 0.04	0.33 +/- 0.01	2.13 +/- 0.06	negative
ElasticNet	0.45 +/- 0.01	2.80 +/- 0.04	0.33 +/- 0.01	2.13 +/- 0.06	negative

REFERENCES

[1] United Nations, "Resolution adopted by the general assembly on 6 july 2017," *United Nations General Assembly*, 2017.

[2] C. Robinson, L. Hou, K. Malkin, R. Soobitsky, J. Czawlytko, B. Dilkina, and N. Jojic, "Large scale high-resolution land cover mapping with multi-resolution data," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[3] C. Robinson, A. Ortiz, K. Malkin, B. Elias, A. Peng, D. Morris, B. Dilkina, and N. Jojic, "Human-machine collaboration for fast land cover mapping," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020.

[4] C. Robinson, F. Hohman, and B. Dilkina, "A deep learning approach for population estimation from satellite imagery," in *Proceedings of the 1st ACM SIGSPATIAL Workshop on Geospatial Humanities*, ACM, 2017, pp. 47–54.

[5] C. Robinson and B. Dilkina, "A machine learning approach to modeling human migration," in *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, ACM, 2018, p. 30.

[6] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," in *Advances in neural information processing systems*, 2017, pp. 3391–3401.

[7] A. Maharana and E. O. Nsoesie, "Use of deep learning to examine the association of the built environment with prevalence of neighborhood adult obesity," *JAMA network open*, vol. 1, no. 4, e181535–e181535, 2018.

[8] N. Abdur Rehman, U. Saif, and R. Chunara, "Deep landscape features for improving vector-borne disease prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 44–51.

[9] S. Piaggesi, L. Gauvin, M. Tizzoni, C. Cattuto, N. Adler, S. Verhulst, A. Young, R. Price, L. Ferres, and A. Panisson, "Predicting city poverty using satellite imagery," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 90–96.

[10] M. Lenormand, S. Huet, F. Gargiulo, and G. Deffuant, "A universal model of commuting networks," *PloS one*, vol. 7, no. 10, e45985, 2012.

[11] M. Lenormand, A. Bassolas, and J. J. Ramasco, "Systematic comparison of trip distribution laws and models," *Journal of Transport Geography*, vol. 51, pp. 158–169, 2016.

[12] S. Arnold, J. Chen, and O. Eggers, "Global and complementary (non-authoritative) geospatial data for sdgs: Role and utilisation," 2019.

[13] UN General Assembly, "Global indicator framework for the sustainable development goals and targets of the 2030 agenda for sustainable development," A/RES/71/313, Revision 2019., Tech. Rep., 2017.

[14] Chesapeake Conservancy, *Land cover data project*, chesapeakeconservancy.org, Ed., `https://chesapeakeconservancy.org/wp-content/uploads/2017/01/LandCover101Guide.pdf`, 2017.

[15] J. J. Feddema, K. W. Oleson, G. B. Bonan, L. O. Mearns, L. E. Buja, G. A. Meehl, and W. M. Washington, "The importance of land-cover change in simulating future climates," *Science*, vol. 310, no. 5754, pp. 1674–1678, 2005.

[16] J Cihlar, "Land cover mapping of large areas from satellites: Status and research priorities," *International journal of remote sensing*, vol. 21, no. 6-7, pp. 1093–1114, 2000.

[17] J. F. Long, "Postcensal population estimates: States, counties, and places," in *Indirect Estimators in US Federal Programs*, Springer, 1996, pp. 59–82.

[18] D. Balk, U Deichmann, G Yetman, F Pozzi, S. Hay, and A Nelson, "Determining global population distribution: Methods, applications and data," *Advances in parasitology*, vol. 62, pp. 119–156, 2006.

[19] M. C. Hansen, P. V. Potapov, R. Moore, M. Hancher, S. Turubanova, A. Tyukavina, D. Thau, S. Stehman, S. Goetz, T. Loveland, *et al.*, "High-resolution global maps of 21st-century forest cover change," *Science*, vol. 342, no. 6160, pp. 850–853, 2013.

[20] H. Zhang, Z.-f. Qi, X.-y. Ye, Y.-b. Cai, W.-c. Ma, and M.-n. Chen, "Analysis of land use/land cover change, population shift, and their effects on spatiotemporal patterns of urban heat islands in metropolitan shanghai, china," *Applied Geography*, vol. 44, pp. 121–133, 2013.

[21] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, and R. Raskar, "Deepglobe 2018: A challenge to parse the earth through satellite images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[22] A. Rakhlin, O. Neuromation, A. Davydow, and S. Nikolenko, "Land cover classification from satellite imagery with u-net and lovász-softmax loss," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 262–266.

[23] *Dstl satellite imagery feature detection*, [Online], 2017.

[24] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, ACM, 2010, pp. 270–279.

[25] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land use classification in remote sensing images by convolutional neural networks," *arXiv preprint arXiv:1508.00092*, 2015.

[26] *ISPRS 2D semantic labeling dataset*, `http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html`.

[27] M. Mahdianpari, B. Salehi, M. Rezaee, F. Mohammadimanesh, and Y. Zhang, "Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery," *Remote Sensing*, vol. 10, no. 7, p. 1119, 2018.

[28] R. Gaetano, D. Ienco, K. Ose, and R. Cresson, "A two-branch cnn architecture for land cover classification of pan and ms imagery," *Remote Sensing*, vol. 10, no. 11, p. 1746, 2018.

[29] M. M. Hayes, S. N. Miller, and M. A. Murphy, "High-resolution landcover classification using random forest," *Remote sensing letters*, vol. 5, no. 2, pp. 112–121, 2014.

[30] T. Kavzoglu, "Object-oriented random forest for high resolution land cover mapping using quickbird-2 imagery," in *Handbook of neural computation*, Elsevier, 2017, pp. 607–619.

[31] X. Li and G. Shao, "Object-based land-cover mapping with high resolution aerial photography at a county scale in midwestern usa," *Remote Sensing*, vol. 6, no. 11, pp. 11 372–11 390, 2014.

[32] X. Li, S. W. Myint, Y. Zhang, C. Galletti, X. Zhang, and B. L. Turner II, "Object-based land-cover classification for metropolitan phoenix, arizona, using aerial photography," *International Journal of Applied Earth Observation and Geoinformation*, vol. 33, pp. 321–330, 2014.

[33] B. Pengra, J. Long, D. Dahal, S. V. Stehman, and T. R. Loveland, "A global reference database from very high resolution commercial satellite data and methodology for application to landsat derived 30m continuous field tree cover data," *Remote Sensing of Environment*, vol. 165, pp. 234–248, 2015.

[34] K. Malkin, C. Robinson, L. Hou, R. Soobitsky, J. Czawlytko, D. Samaras, J. Saltz, L. Joppa, and N. Jojic, "Label super-resolution networks," in *International Conference on Learning Representations (ICLR)*, 2019.

[35] C. Homer, J. Dewitz, L. Yang, S. Jin, P. Danielson, G. Xian, J. Coulston, N. Herold, J. Wickham, and K. Megown, "Completion of the 2011 national land cover database for the conterminous united states–representing a decade of land cover change information," *Photogrammetric Engineering & Remote Sensing*, vol. 81, no. 5, pp. 345–354, 2015.

[36] Iowa DNR, *Iowa 2009 land cover data*, geodata.iowa.gov, Ed., [Online], 2009.

[37] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, e3, 2016.

[38] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[39] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, IEEE, 2017, pp. 1175–1183.

[40] F. Seide and A. Agarwal, "CNTK: Microsoft's open-source deep-learning toolkit," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, pp. 2135–2135.

[41] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson, "Random forests for land cover classification," *Pattern Recognition Letters*, vol. 27, no. 4, pp. 294–300, 2006.

[42] M. Belgiu and L. Drăguţ, "Random forest in remote sensing: A review of applications and future directions," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 114, pp. 24–31, 2016.

[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[44] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[45] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 136–144.

[46] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2017, p. 4.

[47] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *International Conference of Computer Vision (ICCV)*, 2017.

[48] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *European Conference on Computer Vision*, Springer, 2016, pp. 443–450.

[49] C. Tian, C. Li, and J. Shi, "Dense fusion classmate network for land cover classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 192–196.

[50] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison, "Hidden technical debt in machine learning systems," in *Advances in neural information processing systems*, 2015.

[51] T. Griffiths, J. Abbott, and A. Hsu, "Exploring human cognition using large image databases," *Topics in Cognitive Science*, vol. 8, no. 3, 569—588, 2016.

[52] C. Cai, E. Reif, N. Hegde, J. Hipp, B. Kim, D. Smilkov, M. Wattenberg, F. Viegas, G. Corrado, M. Stumpe, and M. Terry, "Human-centered tools for coping with imperfect algorithms during medical decision-making," in *CHI Conference on Human Factors in Computing Systems*, ACM, 2019.

[53] B. Nushi, E. Kamar, and E. Horvitz, "Towards accountable ai: Hybrid human-machine analyses for characterizing system failure," in *AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, AAAI, 2018.

[54] B. Settles, "Active learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 6, no. 1, pp. 1–114, 2012.

[55] C. Zhang, "Active learning and confidence-rated prediction," PhD thesis, UC San Diego, 2017.

[56] C. Zhang, W. Tavanapong, G. Kijkul, J. Wong, P. C. de Groen, and J. Oh, "Similarity-based active learning for image classification under class imbalance," in *2018 IEEE International Conference on Data Mining (ICDM)*, IEEE, 2018, pp. 1422–1427.

[57] T.-K. Huang, L. Li, A. Vartanian, S. Amershi, and J. Zhu, "Active learning with oracle epiphany," in *Advances in Neural Information Processing Systems*, 2016, pp. 2820–2828.

[58] W.-N. Hsu and H.-T. Lin, "Active learning by learning," in *Twenty-Ninth AAAI conference on artificial intelligence*, 2015.

[59] P. Bachman, A. Sordoni, and A. Trischler, "Learning algorithms for active learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017.

[60] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, "Do deep generative models know what they don't know?" In *International Conference on Learning Representations*, 2019.

[61] T.-S. Kuo, K.-S. Tseng, J.-W. Yan, Y.-C. Liu, and Y.-C. F. Wang, "Deep aggregation net for land cover classification," in *Computer Vision and Pattern Recognition (CVPR) workshops*, 2018.

[62] Facebook, *Mapping the world to help aid workers with weakly semi-supervised learning*, `https://ai.facebook.com/blog/mapping-the-world-to-help-aid-workers-with-weakly-semi-supervised-learning/`, 2019.

[63] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Urban land cover classification with missing data modalities using deep convolutional neural networks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 6, pp. 1758–1768, 2018.

[64] M. Schmitt, L. H. Hughes, C. Qiu, and X. X. Zhu, "Sen12ms–a curated dataset of georeferenced multi-spectral sentinel-1/2 imagery for deep learning and data fusion," *arXiv preprint arXiv:1906.07789*, 2019.

[65] P. Y. Simard, S. Amershi, D. M. Chickering, A. E. Pelton, S. Ghorashi, C. Meek, G. Ramos, J. Suh, J. Verwey, M. Wang, *et al.*, "Machine teaching: A new paradigm for building machine learning systems," *arXiv preprint arXiv:1707.06742*, 2017.

[66] X. Zhu, "Machine teaching: An inverse problem to machine learning and an approach toward optimal education," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[67] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" In *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[68] V. Dumoulin, E. Perez, N. Schucher, F. Strub, H. d. Vries, A. Courville, and Y. Bengio, "Feature-wise transformations," *Distill*, 2018, https://distill.pub/2018/feature-wise-transformations.

[69] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," *arXiv preprint arXiv:1610.07629*, 2016.

[70] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[71] Y. Wu and K. He, "Group normalization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[72] B. Settles, "From theories to queries: Active learning in practice," in *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, I. Guyon, G. Cawley, G. Dror, V. Lemaire, and A. Statnikov, Eds., ser. Proceedings of Machine Learning Research, vol. 16, Sardinia, Italy: PMLR, 2011, pp. 1–18.

[73] Chesapeake Conservancy, *High resolution lulc classification accuracy assessment methodology*, chesapeakebay.net, Ed., `https://www.chesapeakebay.net/channel_files/24793/lulcaccuracyassessment_detailed_methodology.pdf`, 2016.

[74] Y. Ma, R. Nowak, P. Rigollet, X. Zhang, and X. Zhu, "Teacher improves learning by selecting a training subset," in *International Conference on Artificial Intelligence and Statistics*, 2018, pp. 1366–1375.

[75] R. Miotto, L. Li, B. A. Kidd, and J. T. Dudley, "Deep patient: An unsupervised representation to predict the future of patients from the electronic health records," *Scientific reports*, vol. 6, p. 26 094, 2016.

[76] J. Chorowski, R. J. Weiss, S. Bengio, and A. v. d. Oord, "Unsupervised speech representation learning using wavenet autoencoders," *arXiv preprint arXiv:1901.08810*, 2019.

[77] K. Aggarwal, S. Joty, L. Fernandez-Luque, and J. Srivastava, "Adversarial unsupervised representation learning for activity time-series," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 834–841.

[78] N. Jean, S. Wang, A. Samar, G. Azzari, D. Lobell, and S. Ermon, "Tile2vec: Unsupervised representation learning for spatially distributed data," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3967–3974.

[79] Y. Duan, X. Tao, M. Xu, C. Han, and J. Lu, "Gan-nl: Unsupervised representation learning for remote sensing image classification," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, IEEE, 2018, pp. 375–379.

[80] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717–1724.

[81] V. R. de Sa, "Learning classification with unlabeled data," in *Advances in neural information processing systems*, 1994, pp. 112–119.

[82] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *International Conference on Learning Representations*, 2018.

[83] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1422–1430.

[84] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European Conference on Computer Vision*, Springer, 2016, pp. 69–84.

[85] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European conference on computer vision*, Springer, 2016, pp. 649–666.

[86] ——, "Split-brain autoencoders: Unsupervised learning by cross-channel prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1058–1067.

[87] X. Wang and A. Gupta, "Unsupervised learning of visual representations using videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2794–2802.

[88] I. Misra, C. L. Zitnick, and M. Hebert, "Shuffle and learn: Unsupervised learning using temporal order verification," in *European Conference on Computer Vision*, Springer, 2016, pp. 527–544.

[89] H.-Y. Lee, J.-B. Huang, M. Singh, and M.-H. Yang, "Unsupervised representation learning by sorting sequences," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 667–676.

[90] C. Doersch and A. Zisserman, "Multi-task self-supervised visual learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2051–2060.

[91] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with exemplar convolutional neural networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 9, pp. 1734–1747, 2015.

[92] P. Goyal, D. Mahajan, A. Gupta, and I. Misra, "Scaling and benchmarking self-supervised visual representation learning," *arXiv preprint arXiv:1905.01235*, 2019.

[93] A. YM., R. C., and V. A., "A critical analysis of self-supervision, or what we can learn from a single image," in *International Conference on Learning Representations*, 2020.

[94] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454.

[95] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[96] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 132–149.

[97] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[98] *Chesapeake land cover*, Maryland split., 2019.

[99] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[100] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2012, pp. 1097–1105.

[101] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[102] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[103] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[104] M. Xie, N. Jean, M. Burke, D. Lobell, and S. Ermon, "Transfer learning from deep features for remote sensing and poverty mapping," *arXiv preprint arXiv:1510.00098*, 2015.

[105] N. Jean, M. Burke, M. Xie, W. M. Davis, D. B. Lobell, and S. Ermon, "Combining satellite imagery and machine learning to predict poverty," *Science*, vol. 353, no. 6301, pp. 790–794, 2016.

[106] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sensing*, vol. 7, no. 11, pp. 14 680–14 707, 2015.

[107] K. Nogueira, O. A. Penatti, and J. A. dos Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *Pattern Recognition*, vol. 61, pp. 539–556, 2017.

[108] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, "Deepsat: A learning framework for satellite imagery," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, 2015, p. 37.

[109] A. Albert, J. Kaur, and M. Gonzalez, "Using convolutional networks and satellite imagery to identify patterns in urban environments at a large scale," *arXiv preprint arXiv:1704.02965*, 2017.

[110] P. Doupe, E. Bruzelius, J. Faghmous, and S. G. Ruchman, "Equitable development through deep learning: The case of sub-national population density estimation," in *Proceedings of the 7th Annual Symposium on Computing for Development*, ACM, 2016, p. 6.

[111] S. K. Smith, "Tests of forecast accuracy and bias for county population projections," *Journal of the American Statistical Association*, vol. 82, no. 400, pp. 991–1003, 1987.

[112] J. F. Long and D. B. McMillen, "A survey of census bureau population projection methods," *Climatic Change*, vol. 11, no. 1, pp. 141–177, 1987.

[113] R. C. Schmitt and A. H. Crosetti, "Accuracy of the ratio-correlation method for estimating postcensal population," *Land Economics*, vol. 30, no. 3, pp. 279–281, 1954.

[114] D. A. Swanson and D. M. Beck, "A new short-term county population projection method," *Journal of Economic and Social Measurement*, vol. 20, no. 1, pp. 25–50, 1994.

[115] M. Mather, K. L. Rivers, and L. A. Jacobsen, "The american community survey.," *Population Bulletin*, vol. 60, no. 3, pp. 1–20, 2005.

[116] US Census Bureau, *Design and methodology: American community survey*, 2009.

[117] U. Deichmann, *A review of spatial population database design and modeling*. National Center for Geographic Information and Analysis, 1996.

[118] S. Hay, A. Noor, A Nelson, and A. Tatem, "The accuracy of human population maps for public health application," *Tropical Medicine & International Health*, vol. 10, no. 10, pp. 1073–1086, 2005.

[119] M. F. Goodchild, L. Anselin, and U. Deichmann, "A framework for the areal interpolation of socioeconomic data," *Environment and planning A*, vol. 25, no. 3, pp. 383–397, 1993.

[120] A. Gaughan, F. R. Stevens, C. Linard, N. N. Patel, and A. J. Tatem, "Exploring nationally and regionally defined models for large area population mapping," *International Journal of Digital Earth*, vol. 8, no. 12, pp. 989–1006, 2015.

[121] A. Sorichetta, G. M. Hornby, F. R. Stevens, A. E. Gaughan, C. Linard, and A. J. Tatem, "High-resolution gridded population datasets for latin america and the caribbean in 2010, 2015, and 2020," *Scientific data*, vol. 2, p. 150 045, 2015.

[122] F. R. Stevens, A. E. Gaughan, C. Linard, and A. J. Tatem, "Disaggregating census data for population mapping using random forests with remotely-sensed and ancillary data," *PloS one*, vol. 10, no. 2, e0107042, 2015.

[123] C. Linard, M. Gilbert, and A. J. Tatem, "Assessing the use of global land cover data for guiding large area population distribution modelling," *GeoJournal*, vol. 76, no. 5, pp. 525–538, 2011.

[124] E. Doxsey-Whitfield, K. MacManus, S. B. Adamo, L. Pistolesi, J. Squires, O. Borkovska, and S. R. Baptista, "Taking advantage of the improved availability of

census data: A first look at the gridded population of the world, version 4," *Papers in Applied Geography*, vol. 1, no. 3, pp. 226–234, 2015.

[125] A. Schneider, M. A. Friedl, and D. Potere, "A new map of global urban extent from modis satellite data," *Environmental Research Letters*, vol. 4, no. 4, p. 044 003, 2009.

[126] J. E. Dobson, E. A. Bright, P. R. Coleman, R. C. Durfee, and B. A. Worley, "Landscan: A global population database for estimating populations at risk," *Photogrammetric engineering and remote sensing*, vol. 66, no. 7, pp. 849–857, 2000.

[127] B. Bhaduri, E. Bright, P. Coleman, and J. Dobson, "Landscan," *Geoinformatics*, vol. 5, no. 2, pp. 34–37, 2002.

[128] L. Seirup and G. Yetman, *U.S. Census Grids (Summary File 1), 2000*, Palisades, NY, 2006.

[129] Center for International Earth Science Information Network -. CIESIN -. Columbia University, *U.S. Census Grids (Summary File 1), 2010*, Palisades, NY, 2017.

[130] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[131] F. Chollet *et al.*, *Keras*, `https://github.com/fchollet/keras`, 2015.

[132] G. Dorigo and W. Tobler, "Push-pull migration laws," *Annals of the Association of American Geographers*, vol. 73, no. 1, pp. 1–17, 1983.

[133] A. De Montis, A. Chessa, M. Campagna, S. Caschili, and G. Deplano, "Modeling commuting systems through a complex network analysis: A study of the italian islands of sardinia and sicily," *Journal of Transport and Land Use*, vol. 2, no. 3, 2010.

[134] T. Dinkelman and M. Mariotti, "The long-run effects of labor migration on human capital formation in communities of origin," *American Economic Journal: Applied Economics*, vol. 8, no. 4, pp. 1–35, 2016.

[135] D. Balcan, V. Colizza, B. Gonçalves, H. Hu, J. J. Ramasco, and A. Vespignani, "Multiscale mobility networks and the spatial spreading of infectious diseases," *Proceedings of the National Academy of Sciences*, vol. 106, no. 51, pp. 21 484–21 489, 2009.

[136] A. Sorichetta, T. J. Bird, N. W. Ruktanonchai, E. zu Erbach-Schoenberg, C. Pezzulo, N. Tejedor, I. C. Waldock, J. D. Sadler, A. J. Garcia, L. Sedda, *et al.*, "Map-

ping internal connectivity through human migration in malaria endemic countries," *Scientific Data*, vol. 3, 2016.

[137] G. Fagiolo and M. Mastrorillo, "International migration network: Topology and modeling," *Physical Review E*, vol. 88, no. 1, p. 012 812, 2013.

[138] A. P. Masucci, J. Serras, A. Johansson, and M. Batty, "Gravity versus radiation models: On the importance of scale and heterogeneity in commuting flows," *Physical Review E*, vol. 88, no. 2, p. 022 812, 2013.

[139] G. K. Zipf, "The $p_1\, p_2\, /d$ hypothesis: On the intercity movement of persons," *American Sociological Review*, vol. 11, no. 6, p. 677, 1946.

[140] M. Schneider, "Gravity models and trip distribution theory," *Papers in Regional Science*, vol. 5, no. 1, pp. 51–56, 1959.

[141] E. S. Lee, "A theory of migration," *Demography*, vol. 3, no. 1, pp. 47–57, 1966.

[142] G. L. Clark and K. P. Ballard, "Modeling out-migration from depressed regions: The significance of origin and destination characteristics," *Environment and Planning A*, vol. 12, no. 7, pp. 799–812, 1980.

[143] M. J. Greenwood, "Human migration: Theory, models, and empirical studies*," *Journal of regional Science*, vol. 25, no. 4, pp. 521–544, 1985.

[144] E. Letouzé, M. Purser, F. Rodríguez, M. Cummins, *et al.*, "Revisiting the migration-development nexus: A gravity model approach," *Human Development Research Paper*, vol. 44, 2009.

[145] A. Noulas, S. Scellato, R. Lambiotte, M. Pontil, and C. Mascolo, "A tale of many cities: Universal patterns in human urban mobility," *PloS one*, vol. 7, no. 5, e37027, 2012.

[146] F. Simini, M. C. González, A. Maritan, and A.-L. Barabási, "A universal model for mobility and migration patterns," *Nature*, vol. 484, no. 7392, pp. 96–100, 2012.

[147] S. A. Stouffer, "Intervening opportunities: A theory relating mobility and distance," *American Sociological Review*, vol. 5, no. 6, pp. 845–867, 1940.

[148] Y. Yang, C. Herrera, N. Eagle, and M. C. González, "Limits of predictability in commuting flows in the absence of data for calibration," *Scientific reports*, vol. 4, 2014.

[149] Y. Ren, M. Ercsey-Ravasz, P. Wang, M. C. González, and Z. Toroczkai, "Predicting commuter flows in spatial networks using a radiation model based on temporal ranges," *Nature communications*, vol. 5, 2014.

[150] J. E. Cohen, M. Roig, D. C. Reuman, and C. GoGwilt, "International migration beyond gravity: A statistical model for use in population projections," *Proceedings of the National Academy of Sciences*, vol. 105, no. 40, pp. 15 269–15 274, 2008.

[151] A. Dennett and A. Wilson, "A multilevel spatial interaction modelling framework for estimating interregional migration in europe," *Environment and Planning A*, vol. 45, no. 6, pp. 1491–1507, 2013.

[152] D. P. Faith, P. R. Minchin, and L. Belbin, "Compositional dissimilarity as a robust measure of ecological distance," *Vegetatio*, vol. 69, no. 1-3, pp. 57–68, 1987.

[153] P. Legendre and L. F. Legendre, *Numerical ecology*. Elsevier, 2012, vol. 24.

[154] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, pp. 785–794.

[155] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[156] L. Mason, J. Baxter, P. L. Bartlett, and M. R. Frean, "Boosting algorithms as gradient descent.," in *NIPS*, 1999, pp. 512–518.

[157] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[158] K. Pierce, "SOI migration data. A new approach: Methodological improvements for SOIC's United States population migration data, calendar years 2011-2012," Statistics of Income, Internal Revenue Service, Tech. Rep., 2015.

[159] X. Guo, Y. Yin, C. Dong, G. Yang, and G. Zhou, "On the class imbalance problem," in *Natural Computation, 2008. ICNC'08. Fourth International Conference on*, IEEE, vol. 4, 2008, pp. 192–201.

[160] Ç. Özden, C. R. Parsons, M. Schiff, and T. L. Walmsley, "Where on earth is everybody? the evolution of global bilateral migration 1960–2000," *The World Bank Economic Review*, vol. 25, no. 1, pp. 12–56, 2011.

[161] World Bank, `http://databank.worldbank.org/data/reports.aspx`, 2017.

[162] R. McLeman and B. Smit, "Migration as an adaptation to climate change," *Climatic change*, vol. 76, no. 1-2, pp. 31–53, 2006.

[163] S. Feng, A. B. Krueger, and M. Oppenheimer, "Linkages among climate change, crop yields and mexico–us cross-border migration," *Proceedings of the National Academy of Sciences*, vol. 107, no. 32, pp. 14 257–14 262, 2010.

[164] R. L. Wilby and R. Keenan, "Adapting to flood risk under climate change," *Progress in physical geography*, vol. 36, no. 3, pp. 348–378, 2012.

[165] M. E. Kahn, *Climatopolis: how our cities will thrive in the hotter future*. Basic Books (AZ), 2013.

[166] G. J. Borjas and J. Monras, "The labor market consequences of refugee supply shocks," National Bureau of Economic Research, Tech. Rep., 2016.

[167] K. Gordon, "The economic risks of climate change in the united states," Risky Business, Tech. Rep., 2014.

[168] S. Shayegh, "Outward migration may alter population dynamics and income inequality," *Nature Climate Change*, vol. 7, no. 11, p. 828, 2017.

[169] G. Luber and M. McGeehin, "Climate change and extreme heat events," *American journal of preventive medicine*, vol. 35, no. 5, pp. 429–435, 2008.

[170] T. C. Brown, V. Mahat, and J. A. Ramirez, "Adaptation to future water shortages in the united states caused by population growth and climate change," *Earth's Future*, vol. 7, no. 3, pp. 219–234, 2019.

[171] J. Church, P. Clark, A. Cazenave, J. Gregory, S. Jevrejeva, A. Levermann, M. Merrifield, G. Milne, R. Nerem, P. Nunn, A. Payne, W. Pfeffer, D. Stammer, and A. Unnikrishnan, "Sea Level Change (Chapter 13)," *Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, 2013.

[172] M. Vermeer and S. Rahmstorf, "Global sea level linked to global temperature," *Proceedings of the National Academy of Sciences*, vol. 106, no. 51, pp. 21 527–21 532, 2009.

[173] W. Sweet, R. Kopp, C. Weaver, J Obeyskera, R. Horton, E. Thieler, and C Zervas, "Global and regional sea level rise scenarios for the united states," NOAA/NOS Center for Operational Oceanographic Products and Services, Tech. Rep. NOAA Technical Report NOS CO-OPS 083, 2017.

[174] B. Güneralp,
bibinitperiodI. Güneralp, and Y. Liu, "Changing global patterns of urban exposure to flood and drought hazards," *Global environmental change*, vol. 31, pp. 217–225, 2015.

[175] K Crossett, B Ache, P Pacheco, and K Haber, "National coastal population report, population trends from 1970 to 2020," *National Oceanic and Atmospheric Administration*, vol. 10, 2014.

[176] M. E. Hauer, J. M. Evans, and D. R. Mishra, "Millions projected to be at risk from sea-level rise in the continental united states," *Nature Climate Change*, vol. 6, no. 7, pp. 691–695, 2016.

[177] R. J. Nicholls, "Analysis of global impacts of sea-level rise: A case study of flooding," *Physics and Chemistry of the Earth, Parts A/B/C*, vol. 27, no. 32, pp. 1455–1466, 2002.

[178] F. Willekens, D. Massey, J. Raymer, and C. Beauchemin, "International migration under the microscope," *Science*, vol. 352, no. 6288, pp. 897–899, 2016.

[179] R. Black, W. N. Adger, N. W. Arnell, S. Dercon, A. Geddes, and D. Thomas, "The effect of environmental change on human migration," *Global Environmental Change*, vol. 21, S3–S11, 2011.

[180] E. Piguet, "Linking climate change, environmental degradation, and migration: A methodological overview," *Wiley Interdisciplinary Reviews: Climate Change*, vol. 1, no. 4, pp. 517–524, 2010.

[181] G. Hugo, "Future demographic change and its interactions with migration and climate change," *Global Environmental Change*, vol. 21, S21–S33, 2011.

[182] K. Wyett, "Escaping a rising tide: Sea level rise and migration in kiribati," *Asia & the Pacific Policy Studies*, vol. 1, no. 1, pp. 171–185, 2014.

[183] K. J. Curtis and A. Schneider, "Understanding the demographic implications of climate change: Estimates of localized population predictions under future scenarios of sea-level rise," *Population and Environment*, vol. 33, no. 1, pp. 28–54, 2011.

[184] S. B. Adamo and A. d. Sherbinin, "The impact of climate change on the spatial distribution of populations and migration," in *Population distribution, urbanization, internal migration and development: An international perspective*, United Nations, 2012, ch. 8, pp. 161–195.

[185] A. M. Findlay, "Migrant destinations in an era of environmental change," *Global Environmental Change*, vol. 21, S50–S58, 2011.

[186] B. C. Thiede and C. L. Gray, "Heterogeneous climate effects on human migration in indonesia," *Population and Environment*, vol. 39, no. 2, pp. 147–172, 2017.

[187] M. E. Hauer, "Migration induced by sea-level rise could reshape the us population landscape," *Nature Climate Change*, 2017.

[188] K. F. Davis, A. Bhattachan, P. D'Odorico, and S. Suweis, "A universal model for predicting human migration under climate change: Examining future sea level rise in bangladesh," *Environmental Research Letters*, vol. 13, no. 6, p. 064 030, 2018.

[189] D. Marcy, W. Brooks, K. Draganov, B. Hadley, C. Haynes, N. Herold, J. McCombs, M. Pendleton, S. Ryan, K. Schmid, *et al.*, "New mapping tool and techniques for visualizing sea level rise and coastal flooding impacts," in *Proceedings of the 2011 Solutions to Coastal Disasters Conference, Anchorage, Alaska, June 26 to June 29*, 2011, p. 474.

[190] R. McLeman, "Developments in modelling of climate change-related migration," *Climatic change*, pp. 1–13, 2013.

[191] U.S. Internal Revenue Service, *Tax Stats - Migration Data*, `https://www.irs.gov/uac/soi-tax-stats-migration-data`, 2017.

[192] E. Fussell, K. J. Curtis, and J. DeWaard, "Recovery migration to the city of new orleans after hurricane katrina: A migration systems approach," *Population and environment*, vol. 35, no. 3, pp. 305–322, 2014.

[193] B. C. O'Neill, E. Kriegler, K. Riahi, K. L. Ebi, S. Hallegatte, T. R. Carter, R. Mathur, and D. P. van Vuuren, "A new scenario framework for climate change research: The concept of shared socioeconomic pathways," *Climatic change*, vol. 122, no. 3, pp. 387–400, 2014.

[194] S. Erlander and N. F. Stewart, *The gravity model in transportation analysis: theory and extensions*. Vsp, 1990, vol. 3.

[195] J. M. Keenan, T. Hill, and A. Gumber, "Climate gentrification: From theory to empiricism in miami-dade county, florida," *Environmental Research Letters*, vol. 13, no. 5, p. 054 001, 2018.

[196] S. Hankey and J. D. Marshall, "Impacts of urban form on future us passenger-vehicle greenhouse gas emissions," *Energy Policy*, vol. 38, no. 9, pp. 4880–4887, 2010.

[197] M. A. Brown, F. Southworth, and A. Sarzynski, "The geography of metropolitan carbon footprints," *Policy and Society*, vol. 27, no. 4, pp. 285–304, 2009.

[198] J. Norman, H. L. MacLean, and C. A. Kennedy, "Comparing high and low residential density: Life-cycle analysis of energy use and greenhouse gas emissions," *Journal of urban planning and development*, vol. 132, no. 1, pp. 10–21, 2006.

[199] B. G. Nichols and K. M. Kockelman, "Life-cycle energy implications of different residential settings: Recognizing buildings, travel, and public infrastructure," *Energy Policy*, vol. 68, pp. 232–242, 2014.

[200] S. Guhathakurta and E. Williams, "Impact of urban form on energy use in central city and suburban neighborhoods: Lessons from the phoenix metropolitan region," *Energy Procedia*, vol. 75, pp. 2928–2933, 2015.

[201] P. G. Newman and J. R. Kenworthy, *Cities and automobile dependence: An international sourcebook*. 1989.

[202] D. Brownstone, "Key relationships between the built environment and vmt," *Transportation Research Board*, vol. 7, 2008.

[203] V. M. Garikapati, D. You, W. Zhang, R. M. Pendyala, S. Guhathakurta, M. A. Brown, and B. Dilkina, "Estimating household travel energy consumption in conjunction with a travel demand forecasting model," *Transportation Research Record: Journal of the Transportation Research Board*, (forthcoming), 2017.

[204] W. Zhang, S. Guhathakurta, and C. Ross, "Trends in automobile energy use and ghg emissions in suburban and inner city neighborhoods: Lessons from metropolitan phoenix, usa," *Energy Procedia*, vol. 88, pp. 82–87, 2016.

[205] Committee for the Study on the Relationships Among Development Patterns, Vehicle Miles Traveled, and Energy Consumption, Board on Energy and Environmental Systems, Transportation Research Board, and National Research Council, *Driving and the Built Environment: The Effects of Compact Development on Motorized Travel, Energy Use, and CO2 Emissions – Special Report 298*. National Academies Press, 2010.

[206] U.S. Energy Information Administration, *U.S. Energy Flow, 2015*, `https://www.eia.gov/totalenergy/data/monthly/pdf/flow/total_energy.pdf`, 2016.

[207] *European Commission Buildings*, `https://ec.europa.eu/energy/en/topics/energy-efficiency/buildings`, Accessed July 29, 2017.

[208] L. Oswaldo, D. Urge-Vorsatz, A. Z. Ahmed, H. Akbari, P. Bertoldi, L. F. Cabeza, N. Eyre, A. Gadgil, L. D. Harvey, Y. Jiang, E. Liphoto, S. Mirasgedis, S. Murakami, J. Parikh, C. Pyke, and M. V. Vilarino, "Climate Change 2014: Mitigation of Climate

Change: Working Group III Contribution to the IPCC Fifth Assessment Report," in. Cambridge University Press, 2014, ch. 9, 671–738.

[209] H.-x. Zhao and F. Magoulès, "A review on the prediction of building energy consumption," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 6, pp. 3586–3592, 2012.

[210] U.S. Energy Information Administration, *Commercial Building Energy Consumption Survey*, `https://www.eia.gov/consumption/commercial/`, 2012.

[211] Z. Li, Y. Han, and P. Xu, "Methods for benchmarking building energy consumption against its past or intended performance: An overview," *Applied Energy*, vol. 124, pp. 325–334, 2014.

[212] G. Tardioli, R. Kerrigan, M. Oates, O. James, and D. Finn, "Data driven approaches for prediction of building energy consumption at urban level," *Energy Procedia*, vol. 78, pp. 3378–3383, 2015.

[213] I. Korolija, L. Marjanovic-Halburd, Y. Zhang, and V. I. Hanby, "Uk office buildings archetypal model as methodological approach in development of regression models for predicting building energy consumption from heating and cooling demands," *Energy and Buildings*, vol. 60, pp. 152–162, 2013.

[214] H. A. Nielsen and H. Madsen, "Modelling the heat consumption in district heating systems using a grey-box approach," *Energy and Buildings*, vol. 38, no. 1, pp. 63–71, 2006.

[215] P. A. Mathew, L. N. Dunn, M. D. Sohn, A. Mercado, C. Custudio, and T. Walter, "Big-data for building energy performance: Lessons from assembling a very large national database of building energy use," *Applied Energy*, vol. 140, pp. 85–93, 2015.

[216] R. E. Brown, T. Walter, L. N. Dunn, C. Y. Custodio, P. A. Mathew, and L Berkeley, "Getting real with energy data: Using the buildings performance database to support data-driven analyses and decision-making," in *Proceedings of the ACEEE Summer Study on Energy Efficiency in Buildings*, 2014, pp. 11–49.

[217] F. Boulaire, A. Higgins, G. Foliente, and C. McNamara, "Statistical modelling of district-level residential electricity use in nsw, australia," *Sustainability science*, vol. 9, no. 1, pp. 77–88, 2014.

[218] P Kuusela, I Norros, R Weiss, and T Sorasalmi, "Practical lognormal framework for household energy consumption modeling," *Energy and Buildings*, vol. 108, pp. 223–235, 2015.

[219] C. E. Kontokosta, "Predicting building energy efficiency using new york city benchmarking data," *Proceedings of the 2012 ACEEE Summer Study on Energy Efficiency in Buildings, Washington, DC, American Council for an Energy-Efficient Economy*, 2012.

[220] G. K. Tso and K. K. Yau, "Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks," *Energy*, vol. 32, no. 9, pp. 1761–1768, 2007.

[221] C. Fan, F. Xiao, and S. Wang, "Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques," *Applied Energy*, vol. 127, pp. 1–10, 2014.

[222] L. Wei, W. Tian, E. A. Silva, R. Choudhary, Q. Meng, and S. Yang, "Comparative study on machine learning for urban building energy analysis," *Procedia Engineering*, vol. 121, pp. 285–292, 2015.

[223] M. Yalcintas and U Aytun Ozturk, "An energy benchmarking model based on artificial neural network method utilizing us commercial buildings energy consumption survey (cbecs) database," *International Journal of Energy Research*, vol. 31, no. 4, pp. 412–421, 2007.

[224] B Howard, L Parshall, J Thompson, S Hammer, J Dickinson, and V Modi, "Spatial distribution of urban building energy consumption by end use," *Energy and Buildings*, vol. 45, pp. 141–151, 2012.

[225] New York City Mayor's Office of Sustainability, *Local Law 84 Data Disclosures*, `http://www.nyc.gov/html/gbee/html/plan/ll84_scores.shtml`, 2016.

[226] M Christenson, H Manz, and D Gyalistras, "Climate warming impact on degree-days and building energy demand in switzerland," *Energy Conversion and Management*, vol. 47, no. 6, pp. 671–686, 2006.

[227] M. A. Brown, M. Cox, B. Staver, and P. Baer, "Modeling climate-driven changes in us buildings energy demand," *Climatic Change*, vol. 134, no. 1-2, pp. 29–44, 2016.

[228] M. Ashfaq, D. Rastogi, R. Mei, S.-C. Kao, S. Gangrade, B. S. Naz, and D. Touma, "High-resolution ensemble projections of near-term regional climate over the continental united states," *Journal of Geophysical Research: Atmospheres*, vol. 121, no. 17, pp. 9943–9963, 2016, 2016JD025285.

[229] New York City Department of City Planning, *PLUTO 16v2*, `https://www1.nyc.gov/site/planning/data-maps/open-data.page`, 2016.

[230] O. N. Keene, "The log transformation is special," *Statistics in medicine*, vol. 14, no. 8, pp. 811–819, 1995.

[231] M. Kuhn and K. Johnson, *Applied predictive modeling*. Springer, 2013, vol. 26.

[232] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, Stanford, CA, vol. 14, 1995, pp. 1137–1145.

[233] United States Department of Energy, *Federal Building Energy Use Benchmarking Guidance, August 2014 Update: Use of Energy and Water Efficiency Measures in Federal Buildings (42 U.S.C. § 8253[f])*, https://energy.gov/sites/prod/files/2014/09/f18/benchmarking_guidance08-2014.pdf, 2014.

[234] P. C. Stern, K. B. Janda, M. A. Brown, L. Steg, E. L. Vine, and L. Lutzenhiser, "Opportunities and insights for reducing fossil fuel consumption by households and organizations," *Nature Energy*, vol. 1, p. 16 043, 2016.

[235] Y.-S. Kim and J. Srebric, "Impact of occupancy rates on the building electricity consumption in commercial buildings," *Energy and Buildings*, 2016.

[236] Y. Huang, J.-l. Niu, and T.-m. Chung, "Study on performance of energy-efficient retrofitting measures on commercial building external walls in cooling-dominant cities," *Applied energy*, vol. 103, pp. 97–108, 2013.

[237] T. Hong, M. A. Piette, Y. Chen, S. H. Lee, S. C. Taylor-Lange, R. Zhang, K. Sun, and P. Price, "Commercial building energy saver: An energy retrofit analysis toolkit," *Applied Energy*, vol. 159, pp. 298–309, 2015.

[238] Energy Information Administration, *Monthly Energy Review*, https://www.eia.gov/totalenergy/data/monthly/pdf/sec2_7.pdf, 2017.

[239] *City of Atlanta Adopts Progressive Energy Policy to Tackle Commercial Energy Use*, http://www.atlantaga.gov/index.aspx?page=672&recordid=3498, 2015.